

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Adam Hornáček

## Virtuální šipkovaná pro Android

Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: RNDr. Petr Kučera, Ph.D.

Studijní program: Informatika

Studijní obor: Programování a softwarové systémy

Praha 2016

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne .....

Podpis autora

Název práce: Virtuální šipkovaná pro Android

Autor: Adam Hornáček

Katedra: Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: RNDr. Petr Kučera, Ph.D., Katedra teoretické informatiky a matematické logiky

Abstrakt: Popisovaný projekt si kladie za cieľ vytvoriť aplikáciu pre mobilný systém Android, ktorá bude pomocou rozšírenej reality zobrazovať používateľom cestu prostredníctvom šípok/bodov, ktoré budú určovať smer ďalšieho pokračovania. Aplikácia umožní používateľom vytvárať a zároveň aj prechádzať cesty. Cesta sa môže ďalej vetviť v určitých miestach podľa zadanej odpovede k otázke. K aplikácii bude vytvorená aj serverová časť umožňujúca výmenu údajov medzi viacerými používateľmi. Aplikácia taktiež umožní hru viacerých hráčov, kde používatelia prechádzajú tú istú cestu a snažia sa získať čo najlepší čas.

Klíčová slova: Hra, Rozšírená realita, Navigácia, Android

Title: Arrows following game for Android

Author: Adam Hornáček

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: RNDr. Petr Kučera, Ph.D., Department of Theoretical Computer Science and Mathematical Logic

Abstract: This project aims to create an application for the mobile system Android which with the help of augmented reality will show the user a route consisting of points/arrows, which represent the direction to follow. The application provides the users with possibilities to create or traverse routes. Routes can be branched at certain locations depending on the answer to a question. The application will be supplemented with a server to allow sharing information between multiple users. The application will also present the option to play the game with multiple users where users traverse the same route in order to achieve the best time.

Keywords: Game, Augmented reality, Navigation, Android

Ďakujem RNDr. Petrovi Kučerovi, Ph.D., za rady pri písaní tejto práce a vývoji aplikácie. Taktiež ďakujem Jakubovi Hornáčkovi za grafický návrh mobilnej aplikácie, neoceniteľné rady a podporu.



# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>  | <b>3</b>  |
| 1.1      | Cieľ práce . . . . .                               | 3         |
| 1.2      | Štruktúra práce . . . . .                          | 4         |
| <b>2</b> | <b>Analýza</b>                                     | <b>5</b>  |
| 2.1      | Cesta . . . . .                                    | 5         |
| 2.1.1    | Objekt cesty . . . . .                             | 6         |
| 2.1.2    | Špeciálne typy objektov cesty . . . . .            | 6         |
| 2.1.3    | Vlastnosti objektov cesty . . . . .                | 6         |
| 2.1.4    | Príklad prechádzania cesty . . . . .               | 7         |
| 2.1.5    | Otázka . . . . .                                   | 7         |
| 2.1.6    | Odpoveď . . . . .                                  | 8         |
| 2.1.7    | Nápoveda . . . . .                                 | 8         |
| 2.1.8    | Reprezentácia cesty . . . . .                      | 8         |
| 2.1.9    | Údaje objektu cesty . . . . .                      | 10        |
| 2.2      | Mobilná aplikácia . . . . .                        | 11        |
| 2.2.1    | Prehľad ciest . . . . .                            | 12        |
| 2.2.2    | Vytváranie cesty . . . . .                         | 17        |
| 2.2.3    | Prechádzanie cesty . . . . .                       | 19        |
| 2.3      | Server . . . . .                                   | 24        |
| 2.3.1    | Implementácia servera . . . . .                    | 25        |
| 2.3.2    | Reprezentácia dát pri posielaní cez sieť . . . . . | 25        |
| 2.3.3    | Implementácia webových služieb . . . . .           | 26        |
| 2.3.4    | Perzistencia dát . . . . .                         | 26        |
| <b>3</b> | <b>Užívateľská dokumentácia</b>                    | <b>28</b> |
| 3.1      | Požiadavky na zariadenie . . . . .                 | 28        |
| 3.2      | Inštalácia . . . . .                               | 28        |
| 3.3      | Prehľad ciest . . . . .                            | 28        |
| 3.3.1    | Vysúvacie menu . . . . .                           | 28        |
| 3.3.2    | Zoznam lokálne uložených ciest . . . . .           | 30        |
| 3.4      | Vytváranie cesty . . . . .                         | 31        |
| 3.4.1    | Základné informácie o ceste . . . . .              | 32        |
| 3.4.2    | Zoznam vytvorených objektov cesty . . . . .        | 33        |
| 3.4.3    | Vytváranie objektu cesty . . . . .                 | 34        |
| 3.4.4    | Pridávanie otázky . . . . .                        | 35        |
| 3.4.5    | Vyberanie nápoved . . . . .                        | 36        |
| 3.4.6    | Výber polohy objektu cesty . . . . .               | 36        |
| 3.4.7    | Zobrazenie objektov cesty na mape . . . . .        | 38        |
| 3.5      | Prechádzanie cesty . . . . .                       | 41        |
| 3.5.1    | Detail cesty . . . . .                             | 41        |
| 3.5.2    | Hra . . . . .                                      | 42        |
| 3.5.3    | Koniec hry . . . . .                               | 44        |

|          |   |           |
|----------|---|-----------|
| <b>4</b> | <b>Programátorská dokumentácia</b>      | <b>45</b> |
| 4.1      | Android . . . . .                       | 45        |
| 4.1.1    | Úvod . . . . .                          | 45        |
| 4.1.2    | Problémy – roztrieštenosť . . . . .     | 45        |
| 4.1.3    | Základné komponenty aplikácie . . . . . | 45        |
| 4.2      | Použité knižnice . . . . .              | 46        |
| 4.2.1    | Mobilná aplikácia . . . . .             | 46        |
| 4.2.2    | Server . . . . .                        | 48        |
| 4.3      | Mobilná aplikácia . . . . .             | 48        |
| 4.3.1    | Prepojenie UI komponentov . . . . .     | 48        |
| 4.3.2    | Perzistencia dát . . . . .              | 50        |
| 4.3.3    | Získavanie polohy zariadenia . . . . .  | 51        |
| 4.3.4    | Rozšírená realita . . . . .             | 52        |
| 4.3.5    | Vykresľovanie . . . . .                 | 52        |
| 4.3.6    | Komunikácia so serverom . . . . .       | 53        |
| 4.3.7    | Hra . . . . .                           | 53        |
| 4.4      | Server . . . . .                        | 53        |
|          | <b>Záver</b>                            | <b>56</b> |
|          | <b>Zoznam použitej literatúry</b>       | <b>57</b> |
|          | <b>Zoznam použitých obrázkov</b>        | <b>59</b> |
|          | <b>Zoznam použitých skratiek</b>        | <b>60</b> |
|          | <b>Prílohy</b>                          | <b>61</b> |

# 1. Úvod

V súčasnosti sa tešia hry veľmi veľkej popularite. V rebríčkoch predajnosti aplikácií zastáva táto kategória jednu z najvyšších priečok. Takisto je veľmi známy geocaching, kde sa podľa GPS súradníc hľadajú balíčky. Tento projekt je zjednotenie týchto dvoch kategórií.

Rozšírená realita je relatívne mladý a dynamicky rozvíjajúci sa odbor. Je veľmi vhodné ju implementovať pre moderné mobilné zariadenia, pretože disponujú nutným technickým vybavením, ktorými sú kvalitná kamera, GPS, gyroskop, akcelerometer, magnetický kompas a výkonný procesor. Medzi takéto zariadenia patria aj zariadenia s operačným systémom Android, ktoré pokrývajú väčšinu trhu a sú cenovo dostupné. Preto sú vhodnými kandidátmi pre realizáciu tohto projektu.

## 1.1 Cieľ práce

Virtuálna šípovaná pre Android je aplikácia, ktorá zábavným a interaktívnym spôsobom umožňuje prechádzanie vonkajším prostredím. Veľmi dobrou analógiou využitia aplikácie je hľadanie pokladu. Aplikácia obsahuje miesta na mape, kam by sa mal používateľ dostať, a pri tomto putovaní mu napomáha. Aplikáciu si môžeme jednoducho predstaviť ako nástroj, ktorý nám zobrazuje náš cieľ a ktorý nám pomáha s priestorovou orientáciou pomocou rozšírenej reality.

Miesta na mape však môžu mať medzi sebou vzťahy a týmto spôsobom tvoria zhuk lokácií, ktoré sa nazývajú cestou. Používateľ si môže zvoliť, ktorú cestu chce prechádzať a táto cesta sa pre neho stane aktívnou. Používateľ sa v každom momente snaží dostať práve na jednu lokáciu. Vzťahy medzi jednotlivými miestami sú reprezentované tým, že jedno miesto môže „ukazovať“ na iné a môže to urobiť nasledovnými spôsobmi:

- Súradnicami nasledujúceho miesta.
- K súradniciam je pripojená aj nakreslená šípka, ktorá reprezentuje smer, ktorým by sa mal používateľ vydať.

Takýmto spôsobom by sme dostali cestu, ktorej priechod by bol stále rovnaký.

S niektorými miestami na mape sa však dá spojiť aj otázka a podľa používateľovej odpovede sa mu zobrazia rôzne informácie nasledujúceho miesta. Týmto spôsobom sa cesta môže vetviť, a teda existuje viac spôsobov, ktorými ju môže používateľ prejsť. Otázky môžu mať správne, čiastočne správne a nesprávne odpovede. Tieto odpovede tak môžu mať so sebou spojený trestný čas, ktorý sa používateľovi pričíta k celkovému času, za ktorý prešiel cestu. Tento čas je kritériom, ktoré určuje, kto prešiel danú cestu najrýchlejšie a pravdepodobne aj najlepšou vetvou cesty. Cesta končí, keď sa používateľ dostane do miesta, ktoré „neukazuje“ na žiadne iné.

Aplikácia neobsahuje žiadne predvolené cesty a každá cesta musí byť vytvorená používateľom. K tomuto účelu aplikácia obsahuje modul, ktorý sa snaží komplikované vytváranie cesty rozdeliť do viacerých krokov s intuitívnym zadávaním údajov. Týmto spôsobom by bolo možné prechádzať cestu na zariadení,

kde by bola vytvorená, čo často nemusí byť tvorcov cieľ. Preto aplikácia umožňuje zdieľanie cesty medzi viacerými používateľmi pomocou súborov alebo servera.

## 1.2 Štruktúra práce

Kapitola **Analýza** sa zaoberá rozoberaním problémov, ktoré sa vyskytli pri vývoji, možnosťami ich riešenia a obsahuje zdôvodnenie vybranej implementácie.

Kapitola **Užívateľská dokumentácia** obsahuje obrázky obrazoviek, ich detailnejší popis a vysvetľuje, ako by sa mala mobilná aplikácia používať.

Kapitola **Programátorská dokumentácia** obsahuje popis implementácie a vzťahov medzi objektmi mobilnej aplikácie. Taktiež popisuje implementáciu servera.

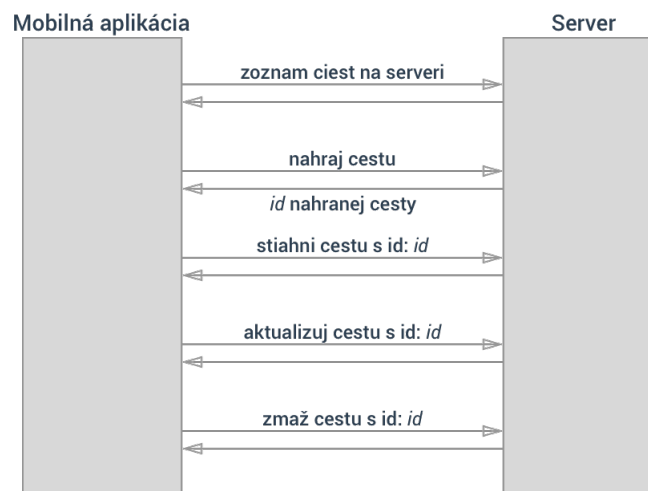
V kapitole **Záver** je popísané celkové zhodnotenie práce. Zhrnutie všetkého, čo bolo implementované a ako by sa to dalo rozšíriť.

## 2. Analýza

Táto kapitola sa venuje problémom spojených s riešením projektu a návrhu ich riešení. Kapitola je rozdelená na nasledovné podkapitoly:

- **2.1 Cesta** – táto podkapitola obsahuje potrebné definície pre správne pochopenie významu cesty v práci a popisuje všetky údaje, ktoré je potrebné mať uložené pre úplný popis cesty.
- **2.2 Mobilná aplikácia** – zameriava sa na časť práce, ktorá bude bežať na mobilnom zariadení.
- **2.3 Server** – zameriava sa na serverovú časť práce.

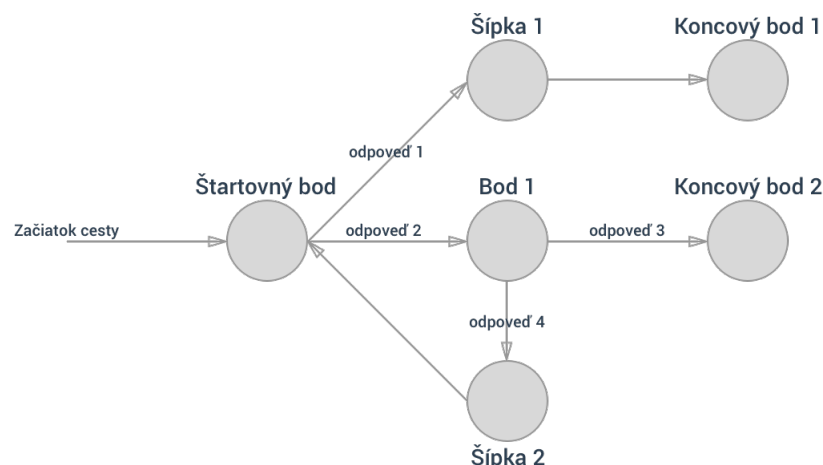
Mobilná aplikácia poskytuje používateľovi množstvo možností. Medzi hlavnú časť patrí aj možnosť vytvorenia cesty. Aby používateľ mohol túto cestu zdieľať elegantným a jednoduchým spôsobom, je v práci integrovaná aj časť programu, ktorá predstavuje server. Používateľ môže na tento server nahrávať svoje vytvorené cesty a taktiež ich aj z neho sťahovať.



Obr. 2.1: Zjednodušená interakcia aplikácie so serverom

### 2.1 Cesta

Cestu si môžeme predstaviť ako orientovaný graf, ktorého jeden vrchol má špeciálny význam a je označený ako štartovný. Tento graf môže byť aj cyklický. Nie je obmedzená jeho veľkosť ani členitosť. Vrcholy tohto grafu reprezentujú objekty cesty. Orientované hrany predstavujú väzby medzi týmito objektmi. Z objektu, z ktorého vychádza orientovaná hrana, sa dá dostať k objektu, do ktorého hrana vchádza. Ak je s objektom cesty zviazaná otázka, tak každá odpoveď na túto otázku predstavuje jednu hranu. Ak s objektom nie je zviazaná otázka, tak z objektu vychádza najviac jedna hrana, ktorá predstavuje prirodzený nasledujúci objekt cesty. Pokiaľ z objektu nevychádza ani jedna hrana, tak je tento objekt koncový a po jeho dosiahnutí hra končí. Hra môže mať viacero koncových objektov. Príklad jednoduchkej cesty je vidieť na obrázku 2.2.



Obr. 2.2: Príklad cesty

### 2.1.1 Objekt cesty

Objekt cesty slúži ako orientačný bod, ktorý používateľovi mobilnej aplikácie napomáha vybrať správny smer pokračovania. Objekt cesty môže byť dvoch typov:

- **Bod** – jednoduchý typ, ktorý predstavuje lokáciu v skutočnom svete, priamo určenú zemepisnou šírkou a zemepisnou dĺžkou, kam by mal užívateľ prísť. S týmto objektom môže byť zviazaná otázka, ktorá umožní vetvenie cesty.
- **Šípka** – zložený typ, ktorý je tvorený svojou lokáciou a orientáciou. Poloha šípky je konštantná a nebude sa meniť podľa polohy používateľa. Orientácia je taktiež konštantná, to znamená, že ak napríklad šípka ukazovala na sever, tak z každého uhla, z ktorého sa na ňu môže používateľ pozerieť, bude stále ukazovať smerom na sever.

### 2.1.2 Špeciálne typy objektov cesty

- **Štartovný objekt cesty** – je to objekt cesty, ktorý reprezentuje začiatok cesty. Pre každú cestu musí byť práve jeden.
- **Koncový objekt cesty** – je to objekt cesty, z ktorého nevychádza žiadna hrana. Musí to byť objekt cesty typu bod, pretože koncový objekt typu šípka nevyhovuje logike prechádzania cesty.

### 2.1.3 Vlastnosti objektov cesty

- S každým objektom cesty môže byť spojených niekoľko nápoved.
- S každým bodom cesty môže byť spojená otázka.
- So žiadnou šípkou nesmie byť spojená otázka. Toto obmedzenie je implementované z dôvodu, že to nedáva z pohľadu prechádzania cesty žiadne výhody, ale pridáva ťažšie pochopenie logiky hry a zložitejšie vytváranie

cesty. Ak by sme šípke povolili otázky, tak s každou odpoveďou by musela byť spojená iná orientácia šípky. Táto istá funkčnosť sa dá dosiahnuť bodom s otázkou, ktorej odpovede sú zviazané s rôznymi šípkami.

### 2.1.4 Príklad prechádzania cesty

Táto sekcia popisuje príklad prechádzania cesty, ktorá je zobrazená na obrázku 2.2. Predpokladajme, že používateľ spustí prechádzanie danej cesty, ďalej môžu nastať nasledovné kroky:

1. Používateľovi sa na obrazovke pomocou rozšírenej reality zobrazí grafický prvok, ktorý reprezentuje miesto štartovného bodu. Používateľ uvidí jeho meno a vzdialenosť od daného objektu.
2. Používateľ príde do okolia štartovného bodu a zobrazí sa mu dialógové okno s otázkou, ktoré bude obsahovať odpovede: odpoveď 1, odpoveď 2.
3. Používateľ vyberie odpoveď 2. Bod 1 sa stane jeho aktívnym bodom, ktorý uvidí na obrazovke.
4. Používateľ sa dostaví do okolia bodu 1, znova mu bude položená otázka a vyberie odpoveď 4.
5. Používateľ opäť uvidí grafický prvok podobný tomu ako pri bode, ale keď sa dostaví do okolia šípky 2, tak sa mu pomocou rozšírenej reality zobrazí šípka, ktorá bude určovať smer jeho ďalšieho pokračovania. Pokiaľ opustí okolie tejto šípky, tak sa mu znova zobrazí grafický prvok, ktorý reprezentuje nasledujúci objekt cesty, v tomto prípade je to štartovný bod.
6. Používateľ sa opäť dostaví do štartovného bodu, teraz zvolí odpoveď 1.
7. Používateľ sa dostaví k šípke 1, ktorej funkčnosť bude veľmi podobná šípke 2. Po opustení jej okolia uvidí polohu koncového bodu 1.
8. Používateľ sa dostaví do okolia koncového bodu 1, čím pre neho prechádzanie cesty končí.

### 2.1.5 Otázka

Otázka predstavuje možnosť vetvenia cesty. Každá otázka je určená textom otázky, typom a jej odpoveďami. Text otázky je text, ktorý sa používateľovi ukáže pri zadaní otázky. Možné typy otázky:

- **Výber z možností** – používateľovi sa zobrazí zoznam všetkých odpovedí a on z nich bude môcť zvoliť jednu odpoveď, ktorú pokladá za správnu. Odpovede sa pri tomto type otázky budú zobrazovať v náhodnom poradí.
- **Písanie odpovede** – používateľovi sa iba zobrazí okno, kam bude musieť napísať svoju odpoveď. Do úvahy sa nebude brať rozdiel vo veľkosti písmen.

### 2.1.6 Odpoved'

Odpoved' urcuje dalsi priebeh cesty. Každá odpoved' má text a urcuje nasledujúci objekt cesty. S každou odpoveďou môže byť spojený trestný čas. Týmto spôsobom môže používateľ získať rozličný čas pri prechádzaní cesty, ktorý slúži ako kritérium porovnávania ich priechodov.

### 2.1.7 Nápoveda

S každým objektom cesty môže byť zviazaných niekoľko nápovedí, ktoré môžu používateľovi pomôcť dostať sa na žiadané miesto. Nápoveda je predstavovaná textom, ktorým môže tvorca cesty používateľovi napomôcť. S každou nápovedou môže byť zviazaný aj trestný čas, ktorý sa pričíta k celkovému času prejdienia cesty.

### 2.1.8 Reprezentácia cesty

Táto podkapitola popisuje údaje zviazané s cestou, ktoré sú spoločné pre mobilnú aplikáciu a server. Špecifické údaje budú popísané v ich príslušných podkapitolách. Je podstatné, aby bola zvolená reprezentácia, ktorá umožňuje nekomplikovanú prácu, vhodné mapovanie do SQL<sup>1</sup> tabuliek a adekvátnu komunikáciu so serverom. Keďže do jedného objektu cesty môže vchádzať viac hrán, tak priame mapovanie pomocou viacnásobnej referencie, ktoré na mobilnom zariadení fungovalo korektne, môže spôsobovať problémy pri konverzii do formátu JSON<sup>2</sup>, ktorý slúži na prenos cesty medzi dvomi zariadeniami, pretože ten istý objekt cesty by tam bol uložený viackrát. Z toho dôvodu obsahuje cesta jednu hashovaciu tabuľku, pomocou ktorej sa pristupuje ku všetkým objektom cesty. Typom mapovaných hodnôt tejto tabuľky bude objekt cesty. Na výber kľúča máme 2 možnosti:

- **Reťazec** – každý bod má svoje vlastné meno, ktoré je unikátne v rámci jednej cesty (kvôli možnosti jednoduchšie medzi nimi rozlíšiť). Reťazec je v Jave nemenný objekt<sup>3</sup>, takže je vhodným kandidátom na kľúč. Nevýhodou je dlhší čas počítania hash kódu a väčšia pamäťová náročnosť pri držaní kľúča.
- **Celé číslo** – ku každému bodu by sme museli pridať atribút, ktorý by reprezentoval jeho unikátne id v rámci jednej cesty. Nevýhodou je nutnosť stĺpca navyše v SQL tabuľke. Pamäťová náročnosť bude pravdepodobne menšia, keďže si budeme v miestach označujúce ďalší bod pamätať číslo a nie reťazec.

**Zvolené riešenie** – najmä z hľadiska efektivity bolo zvolené riešenie, kde kódom hashovacej tabuľky je celé číslo. Tým sa nám naskytla nová možnosť, ako urobiť prístup k objektom cesty ešte efektívnejší. Môžeme objekty cesty reprezentovať ako pole, ale musíme udržiavať identifikátory bodov v rámci jednej cesty od 0 po počet objektov mínus 1. Problém tohto riešenia je iba pri mazaní objektu

<sup>1</sup>Structured Query Language, <https://en.wikipedia.org/wiki/SQL>

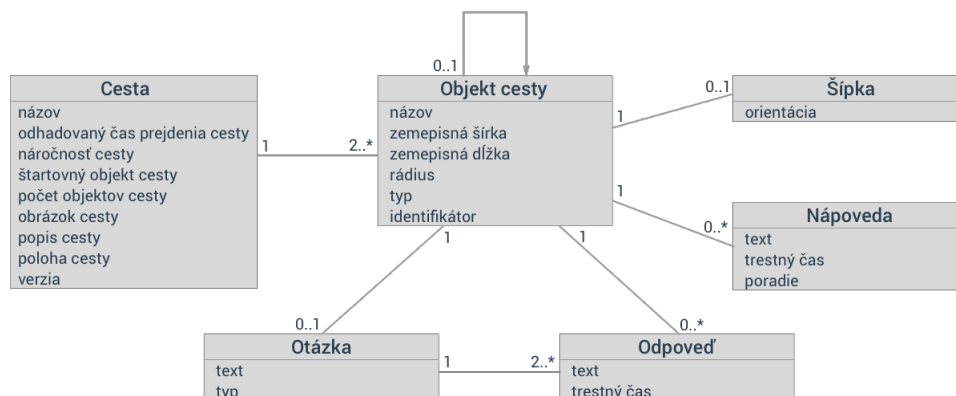
<sup>2</sup>JavaScript Object Notation, <http://www.json.org/>

<sup>3</sup>Immutable object, [https://en.wikipedia.org/wiki/Immutable\\_object](https://en.wikipedia.org/wiki/Immutable_object)



cesty, kedy musíme zmenšiť všetky odkazy na objekty s vyšším identifikátorom ako mazaný objekt o 1.

Cesta bude uložená v databázach, ako aj v serverovej, tak aj v databáze mobilnej aplikácie. Preto musia byť dáta reprezentované pomocou tabuliek. Jednotlivé tabuľky budú mať podobnú štruktúru, ktorá je popísaná na obrázku 2.3.



Obr. 2.3: Znázornenie tabuliek, ktoré obsahujú údaje cesty

### Údaje cesty

- **Názov** – reťazec, ktorý obsahuje krátky názov cesty. Tento reťazec nemusí byť unikátny ani v rámci aplikácie, ani v rámci servera. Slúži len pre jednoduchšiu orientáciu v zozname ciest.
- **Odhadovaný čas prejdenia cesty** – celé číslo, ktoré určuje predpokladaný počet minút trvania cesty určený tvorcom pri jej vytváraní.
- **Náročnosť cesty** – celé číslo, ktoré určuje, aká je cesta náročná.
- **Štartovný objekt cesty** – cudzí kľúč, ktorý reprezentuje objekt cesty určujúci začiatok cesty.
- **Počet objektov cesty** – celé číslo, ktoré určuje celkový počet objektov zviazaných s touto cestou.
- **Obrázok cesty** – binárne dáta, ktoré obsahujú obrázok cesty. Tento obrázok si môže zvoliť tvorca cesty a bude sa zobrazovať v zozname ciest. Pokiaľ nevyberie žiadny obrázok, tak sa v zozname ciest bude zobrazovať predvolený obrázok, ktorý je určený aplikáciou.
- **Popis cesty** – reťazec, ktorý obsahuje dlhší popis cesty.
- **Poloha cesty** – reťazec, ktorý popisuje polohu, kde sa cesta nachádza.
- **Verzia** – celé číslo, ktoré popisuje verziu cesty, aby sme mohli porovnať, či sa na serveri nachádza novšia verzia.

### 2.1.9 Údaje objektu cesty

- **Názov** – reťazec, ktorý obsahuje krátky názov objektu. Tento názov musí byť unikátny v rámci jednej cesty. Zobrazuje sa používateľovi pri prechádzaní cesty.
- **Zemepisná šírka** – reálne číslo, ktoré určuje zemepisnú šírku objektu.
- **Zemepisná dĺžka** – reálne číslo, ktoré určuje zemepisnú dĺžku objektu.
- **Rádus** – celé číslo určujúce počet metrov, ktoré tvoria polomer okolo objektu. Keď používateľ do tohto polomeru vojde, tak sa mu daný objekt zaráta ako prejdenný.
- **Typ** – celé číslo, ktoré určuje typ objektu, tzn. či je to bod alebo šípka a či obsahuje otázku.
- **Identifikátor** – celé číslo, ktoré je unikátne v rámci jednej cesty a slúži pre identifikáciu objektu v ceste.

#### Údaje o šípke

- **Orientácia** – celé číslo, ktoré určuje azimut resp. otočenie šípky voči severu.

#### Údaje o otázke

- **Text** – reťazec obsahujúci text otázky, ktorý sa zobrazí používateľovi pri zobrazení otázky.
- **Typ** – celé číslo, ktoré určuje, či sa na otázku odpovedá výberom z odpovedí alebo priamo vpisovaním odpovede.

#### Údaje o odpovedi

- **Text** – reťazec určujúci text odpovede, ktorá sa podľa typu otázky buď zobrazí ako možnosť, alebo sa bude porovnávať s vpísanou odpoveďou.
- **Trestný čas** – celé číslo určujúce počet minút, ktoré sa používateľovi pričítajú k celkovému času po vybratí tejto odpovede.

#### Údaje o nápovede

- **Text** – reťazec obsahujúci text, ktorý sa zobrazí používateľovi po vyžiadaní nápovedy k objektu cesty.
- **Trestný čas** – celé číslo určujúce počet minút, ktoré sa používateľovi pričítajú k celkovému času po zobrazení tejto nápovedy.
- **Poradie** – celé číslo, ktoré určuje poradie nápovedy.

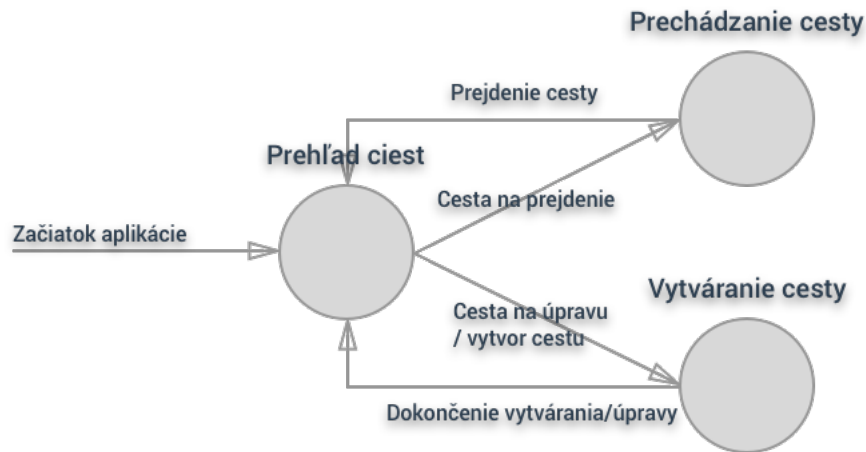
## Popísanie vzťahov medzi jednotlivými tabuľkami

- **Cesta – Objekt cesty** – každá cesta sa musí skladať z aspoň dvoch objektov cesty a každý objekt cesty patrí práve jednej ceste.
- **Objekt cesty – Šípka** – každý objekt cesty môže byť typu šípka, potom potrebujeme dodatočné informácie, ktoré sú uložené v tabuľke Šípka. Tieto dodatočné informácie sa viažu k práve jednému objektu cesty.
- **Objekt cesty – Nápoveda** – každý objekt cesty môže mať so sebou zviazaný ľubovoľný počet nápoved, pričom každá nápoveda sa viaže k práve jednému objektu cesty.
- **Objekt cesty – Otázka** – ku každému objektu cesty môže byť priradená otázka, ktorá je však priradená iba k tomuto jednému objektu.
- **Otázka – Odpoveď** – s každou otázkou musia byť zviazané aspoň dve odpovede, ktoré sú súčasťou iba tejto jednej otázky.
- **Odpoveď – Objekt cesty** – každá odpoveď musí ukazovať na jeden objekt cesty, ktorý sa stane nasledujúcim po odpovedaní tejto otázky. Každý objekt cesty môže byť zviazaný s ľubovoľným počtom odpovedí. Pokiaľ odpoveď bude mať prázdny text, tak táto odpoveď bude mať špeciálny význam a bude reprezentovať predvolený nasledujúci objekt pri napísaní odpovede, ktorá sa nezhoduje so žiadnou zo zadaných odpovedí.
- **Objekt cesty – Objekt cesty** – predvolený nasledujúci objekt cesty, pokiaľ s ním nebola zviazaná žiadna otázka. Každý objekt cesty môže mať maximálne jeden predvolený nasledujúci objekt cesty.

## 2.2 Mobilná aplikácia

Aplikácia sa bude skladať z 3 hlavných modulov:

- **Prehľad ciest** – vstupný bod aplikácie. V tomto module bude zobrazený zoznam ciest, ktoré sú dostupné na prechádzanie. Tento modul bude taktiež obsahovať možnosti, pomocou ktorých sa aplikácia bude môcť prepnúť do modulu vytvárania cesty. Pokiaľ sa modulu vytvárania cesty predá cesta ako parameter, tak sa budú vykonávať jej úpravy. Používateľ bude schopný taktiež vybrať cestu, ktorú by chcel prejsť, a aplikácia sa prepne do modulu prechádzania cesty.
- **Vytváranie cesty** – v tomto module bude používateľ schopný vytvoriť alebo upraviť plnohodnotnú cestu. Po dokončení vytvárania/úpravy sa aplikácia vráti do modulu prehľadu ciest, kde bude novovytvorená/upravená cesta ihneď dostupná všetkým funkciám modulu prehľadu ciest.
- **Prechádzanie cesty** – tento modul dostane ako parameter cestu, ktorú chce používateľ prejsť. Po dokončení prechádzania cesty sa aplikácia vráti späť do modulu prehľadu ciest.



Obr. 2.4: Interakcia modulov

Zjednodušený model interakcie medzi jednotlivými modulmi je vidieť na obrázku 2.4.

Mobilná aplikácia si musí o ceste pamätať niekoľko údajov navyše od tých, ktoré sme popísali v predchádzajúcej kapitole. Týmito údajmi sú:

- **Id cesty na serveri** – celé číslo, ktoré identifikuje cestu na serveri. Táto informácia je potrebná pre aktualizáciu a mazanie cesty.
- **Údaj, či má používateľ právo cestu upravovať** – pokiaľ bola cesta vytvorená na danom zariadení, tak ju má používateľ možnosť upravovať. Bez ohľadu na to, ako je identifikovaný. Používateľ bude môcť taktiež upravovať cesty, ktoré sám nahral na server.

### 2.2.1 Prehľad ciest

Prehľad ciest je vstupným bodom celej mobilnej aplikácie. Účelom tohto modulu je sprístupniť používateľovi cesty, ktoré sú uložené lokálne na zariadení, ale aj vzdialene na serveri. Modul taktiež umožňuje základnú manipuláciu s cestami. Modul obsahuje nasledovné funkcie:

- **Zoznam lokálnych ciest** – hlavná časť tohto modulu, ktorá je predvolenou a zobrazí sa vždy ako prvá pri vstupe do tohto modulu. Na celej obrazovke bude zobrazený zoznam, ktorého každá položka bude reprezentovať jednu cestu lokálne uloženú na danom zariadení. Každá položka by mala zobrazovať obrázok cesty a všetky dôležité informácie o ceste, ktoré používateľovi poskytnú možnosť jednoducho rozlíšiť medzi väčším počtom položiek.
- **Zoznam ciest uložených na serveri** – podobný vzhľad a funkčnosť ako pri zozname lokálnych ciest, ale informácie sa budú získavať zo vzdialeného servera namiesto z lokálnej databázy. Zo servera sa stiahnu iba základné informácie o cestách a zobrazia sa ako položky zoznamu. V zozname zatiaľ nebude možné vyhľadávať kvôli predpokladu, že to zatiaľ nebude nutné. Pokiaľ sa vyskytne táto potreba, tak sa táto možnosť doimplementuje v novších verziách.

- **Vytvorenie novej cesty** – prejde do modulu vytvárania cesty. Po úspešnom vytvorení sa aplikácia vráti späť na zoznam lokálnych ciest, ktorý už bude obsahovať novovytvorenú cestu.
- **Úprava cesty** – používateľ bude schopný označiť cestu zo zoznamu lokálnych ciest, pričom sa mu spustí modul vytvárania cesty, ktorému predáme ako parameter cestu, ktorú chceme upraviť.
- **Nahratie cesty na server** – modul poskytne používateľovi možnosť označiť jednu cestu zo zoznamu lokálnych ciest a nahráť túto cestu na server.
- **Zmazanie existujúcej cesty** – používateľ bude mať možnosť zmazať cestu, ktorá je lokálne uložená na zariadení, alebo cestu, ktorú sám nahral na server.
- **Prejdenie do modulu prechádzania cesty** – modul umožní vybrať cestu zo zoznamu a začať ju prechádzať.
- **Stiahnutie cesty zo servera** – modul umožní cestu stiahnuť zo servera a predá ju do modulu prechádzania cesty s oznámením, že sa jedná o cestu stiahnutú z internetu, aby ďalší modul vedel ponúknuť možnosť uloženia cesty do lokálnej databázy.
- **Načítanie ciest z externého úložiska** – používateľ bude schopný načítať cestu do lokálnej databázy zo súboru, ktorý je uložený na externom úložisku.
- **Zmeniť nastavenia** – modul umožní zmeniť niektoré nastavenia aplikácie.
- **Informácie** – modul zobrazí používateľovi základné informácie o projekte, jeho tvorcach a za akým účelom bol projekt vytvorený.
- **Prihlásenie pomocou účtov** – možnosť prihlásenia pre potreby nahrávania ciest na server a hrania hier viacerých hráčov.
- **Odhlásenie z účtov** – možnosť odhlásenia z účtov. Po odhlásení nebudú služby, pre ktoré je potrebné prihlásenie, funkčné.
- **Zobrazenie pozvánok do hry viacerých hráčov** – rýchly prístup k pozvánkam od ostatných hráčov. Používateľ bude môcť pozvánky prijať alebo odmietnuť. Po prijatí pozvánky sa pošle daná pozvánka do modulu prechádzania cesty.
- **Zobrazenie bonusov** – zobrazí používateľovi obrazovku bonusov, kde uvidí, ktoré bonusy už získal a ktoré je ešte možné získať.
- **Synchronizácia ciest** – modul umožní používateľovi synchronizovať cesty so serverom.

## Exportovanie cesty do súboru

Táto funkcia sa spýta používateľa na meno súboru a uloží cestu do súboru s daným názvom na primárne zdieľané/externé úložisko<sup>4</sup> do adresára Arrows s príponou .arrows. Cestu budeme reprezentovať ako jednoduchý textový súbor vo formáte JSON, ktorý bude obsahovať obrázok cesty zakódovaný vo formáte Base64 a cestu. Avšak z praktického dôvodu, aby nemohli používatelia jednoduchým prečítaním súboru zistiť odpovede, a teda aj najlepšiu cestu k víťazstvu, bude cesta zašifrovaná. Zatiaľ nebude možné uložiť cestu v nezašifrovanom stave. Cestu budeme šifrovať pomocou AES<sup>5</sup> šifry, známou tiež podľa mena Rijndael, ktorá je symetrickou šifrou a pokladaná za štandard podľa NIST<sup>6</sup>.

## Nastavenia

Používateľ bude schopný zmeniť nasledovné funkcie aplikácie:

- **Farba šípky** – používateľ bude môcť zmeniť farbu šípky.
- **Čas obnovy polohy** – rýchlejší čas obnovy polohy zlepšuje herný zážitok, ale znižuje výdrž batérie. Používateľ bude môcť vybrať jednu z nasledovných možností:
  - **Rýchly** – aktualizuje polohu každých 5 sekúnd.
  - **Stredný** – aktualizuje polohu každých 10 sekúnd.
  - **Pomalý** – aktualizuje polohu každých 20 sekúnd.

## Prihlásenie pomocou účtov

Bude možné sa prihlásiť pomocou dvoch účtov, pričom každý poskytuje inú funkcionálnosť. Týmto účtami sú:

### Účet pre manipuláciu s cestami na serveri

Keďže aplikácia poskytuje možnosť nahráť cestu na server, je nutné aby sme vedeli identifikovať používateľa, ktorý tak vykonal. Táto identifikácia zviaže používateľa s nahranou cestou a okrem nahrávania cesty na server mu poskytne aj možnosti, medzi ktoré patria:

- **Mazanie cesty zo servera** – používateľ sa môže rozhodnúť zmazať cestu zo servera. Týmto sa zmažú všetky údaje o danej ceste na serveri a už ju nebude možné z neho stiahnuť. Cesta však bude naďalej dostupná u všetkých používateľov, ktorí už danú cestu majú uloženú na zariadení.
- **Aktualizácia aktuálnej cesty** – používateľ môže upraviť už vytvorenú cestu. Ak upraví cestu, ktorá je už nahraná na serveri, tak sa aplikácia automaticky pokúsi o jej aktualizáciu aj na serveri. Ak to nebude možné,

---

<sup>4</sup>tradične je to SD karta, ale nemusí to tak byť

<sup>5</sup>Advanced Encryption Standard, [https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)

<sup>6</sup>National Institute of Standards and Technology, [https://en.wikipedia.org/wiki/National\\_Institute\\_of\\_Standards\\_and\\_Technology](https://en.wikipedia.org/wiki/National_Institute_of_Standards_and_Technology)

napr. z dôvodu nepripojenia na internet, tak sa bude používateľ môcť pokúsiť o manuálnu aktualizáciu neskôr.

Práve kvôli týmto možnostiam musíme uchovávať citlivé informácie o používateľoch, čím sa kladie veľký dôraz na bezpečnosť. K implementácii účtu môžeme pristúpiť dvomi rôznymi spôsobmi:

- **Vlastný návrh** – výhodou tohto riešenia je, že všetko je navrhnuté podľa potrieb aplikácie a používajú sa iba údaje, ktoré sú naozaj potrebné. Nevýhodou sú možné bezpečnostné riziká a ťažšia implementácia. Takisto nutnosť vytvorenia modulov, kde by sa mohol používateľ registrovať, prihlasovať a meniť svoje osobné údaje.
- **Google účet** – nielen aplikácie, ale aj mnoho známych webových portálov poskytuje možnosť prihlásenia pomocou Google účtu. Táto možnosť je pre veľa používateľov veľmi príjemná, pretože si nemusia pamätať rôzne prihlasovacie údaje a prihlásenie je často automatické, ak máte integrovaný Google účet, čo väčšina používateľov s mobilnými zariadeniami Android má. Výhodou je dobrá podpora, bezpečnosť a rozšírenosť tejto možnosti.

**Zvolené riešenie** – nakoniec bola zvolená implementácia pomocou Google účtu. Najmä kvôli menšej pravdepodobnosti výskytu chýb a ich rýchlejšie opravenie zo strany Googlu. V implementácii bude pridané tlačidlo, pomocou ktorého sa používateľovi zobrazí prihlasovacia časť. Tá zobrazí na výber už integrované Google účty na zariadení alebo možnosť vytvoriť nový účet.

### Účet pre hranie hry viacerých hráčov

Aplikácia taktiež umožňuje používateľom hrať hru viacerých hráčov, takže musíme používateľov nejako identifikovať a sprostredkovať medzi nimi komunikáciu. Možnými riešeniami sú:

- **Vlastný návrh** – aplikácia by komunikovala so serverom a pomocou neho s ostatnými klientmi. Návrh a zabezpečenie by neboli triviálne a pridáva priestor na množstvo chýb. Výhodou je, že riešenie by priamo vyhovovalo potrebám aplikácie.
- **Google Play Games Services [1]** – poskytujú Real-time Multiplayer, ktorý zabezpečuje hru viacerých hráčov. Je napojený priamo na aplikáciu Google hry a vytvára peer-to-peer komunikáciu medzi klientmi. Umožňuje pozývať priateľov do hry a taktiež nájsť používateľov v blízkosti. Nevýhodou je nutnosť mať nainštalovanú aplikáciu Google hry na zariadení a byť do nej prihlásený.

**Zvolené riešenie** – najmä vďaka integrácii do systému Android považujem Google Play Games Services za lepšie a efektívnejšie riešenie. Takisto združuje výsledky z mnohých aplikácií do jednej, čím zjednodušuje nutnosť používateľov pamätať si mnoho údajov a prihlasovať sa viacerými účtami.

Z dôvodu, že Google nepodporuje prihlásenie pomocou Google Sign In a Google Games zároveň, je nutné implementovať tlačidlá na prihlasovanie pre každú službu zvlášť.

## Načítanie ciest z externého úložiska

Po kliknutí na toto tlačidlo máme viac možností ako načítať súbor na naše zariadenie, aby bol dostupný v prehľade ciest. Tieto možnosti sú:

- **Spýtať sa používateľa na cestu k súboru** – veľmi jednoduché a priamočiare riešenie, ale nie je veľmi používateľsky prívetivé, pretože používateľ by si musel pamätať alebo niekde zapísať presnú absolútnu cestu k súboru.
- **Prejsť celé externé úložisko** – veľmi jednoduché pre používateľa, ale veľmi neefektívne, pretože by sme museli prijímať len súbory s určitou koncovkou (kvôli orezaniu) a pre každý takýto súbor kontrolovať, či už náhodou nebol pridaný do lokálnej databázy.
- **Implementovať vlastný jednoduchý file manager** – zobrazíť používateľovi jednoduchý zoznam adresárov a súborov. Po kliknutí na súbor ho načítať a po kliknutí na adresár sa doň vnoriť. Nevýhodou je nutnosť implementovať niečo, čo používateľ už pravdepodobne má nainštalované na zariadení.
- **Spustiť Android Intent na vybratie súboru** – spustí Android aplikáciu, z ktorej si budeme môcť zvoliť náš obľúbený file manager, z ktorého si budeme môcť vybrať cestu. Veľkými výhodami sú integrácie napr. Dropboxu alebo Google Drive. Nevýhodou je nutnosť mať nainštalované tieto aplikácie a nejaký file manager.
- **Použiť knižnicu na prechádzanie súborov** – použili by sme knižnicu, ktorá nám poskytne zjednodušený file manager. Hlavnou nevýhodou by bol iný dizajn používateľského rozhrania a znova reimplementácia už pravdepodobne nainštalovanej aplikácie.

**Zvolené riešenie** – nakoniec bolo zvolené riešenie, kde sa spustí Android Intent na vybratie súborov pre jednoduchosť implementácie a dobrej integrácii.

## Synchronizácia ciest

V prípade, že by používateľ prešiel na nové zariadenie, zmazal lokálnu databázu aplikácie, odinštaloval aplikáciu alebo vykonal factory reset, je nutné, aby si mohol všetky svoje už vytvorené cesty, ktoré sú uložené na serveri, opäť stiahnuť do aplikácie a mohol ich naďalej upravovať. Preto aplikácia poskytne používateľovi možnosť synchronizovať cesty so serverom, čo porovná všetky jeho lokálne uložené cesty s cestami, ktoré má uložené na serveri a pokiaľ niektoré nebudú uložené na zariadení, tak sa stiahnu a uložia do lokálnej databázy.

Pokiaľ však nenastanú vyššie uvedené prípady, tak všetky zmeny, ktoré používateľ vykoná na zariadení, sa pokúsia vykonať aj na serveri. Pokiaľ používateľ zmaže cestu, tak sa zmaže aj na serveri. Pokiaľ používateľ nebude mať internetové pripojenie, tak ho aplikácia upozorní, že cesta bude stále dostupná na serveri. Pokiaľ používateľ vykoná zmeny a uloží ich, tak sa zvýši verzia cesty a aplikácia sa pokúsi aktualizovať cestu aj na serveri. Pokiaľ používateľ nebude mať internetové pripojenie, tak bude musieť vykonať aktualizovanie cesty manuálne neskôr.



Pokiaľ bude mať používateľ stiahnutú cestu zo servera alebo načítanú zo súboru, nebude môcť na nej vykonávať žiadne zmeny. Pokiaľ bude cesta, ktorú používateľ stiahol z internetu, aktualizovaná jej tvorcom, synchronizácia aktualizuje túto cestu na najnovšiu verziu.

## 2.2.2 Vytváranie cesty

Jedným z hlavných prvkov aplikácie je možnosť vytvárať cesty. Keďže cesta môže byť veľmi komplikovaná, je veľmi dôležité, aby jej vytváranie bolo jednoduché a intuitívne. Máme na výber z viacerých možností, ako vytváranie implementovať:

- **Súbor špeciálneho formátu** – táto možnosť je veľmi nepraktická, pretože písanie dlhších textov je na mobilných zariadeniach s dotykovou obrazovkou nepohodlné a zdĺhavé. Takže by museli byť cesty tvorené na počítači a potom kopírované na zariadenie. Hlavnou výhodou tohto riešenia je jednoduchosť implementácie avšak za cenu nekvalitného používateľského zážitku.
- **Vytváranie priamo v aplikácii zadávaním presných údajov** – používateľ by bol nútený zadávať do aplikácie presné GPS súradnice, ktoré sú reprezentované reálnymi číslami s desatinným rozvojom približne 15 miest. Používateľ by sa mohol ľahko pomýliť a taktiež by musel získavať tieto informácie z iných zdrojov. Hlavnou nevýhodou je ťažšia implementácia.
- **Vytváranie priamo v aplikácii vyberaním bodov z mapy** – aplikácia by zobrazila interaktívnu mapu, v ktorej by sa mohol používateľ orientovať a klikaním na určité miesto by sa získali presné súradnice. Nevýhodou je opäť ťažšia implementácia, nemožnosť zadať presné súradnice a nutnosť trafenia sa presne na mapu.
- **Vytváranie pomocou webovej aplikácie** – používateľ by bol schopný vytvárať cestu pomocou webového prehliadača na osobnom počítači. Keďže je k dispozícii väčšia obrazovka, bolo by možné zobraziť viac informácií naraz a vytváranie by bolo kompaktnejšie. Bola by nutnosť prihlasovania pomocou rovnakého účtu ako na zariadení a používateľ by si musel cestu stiahnuť na zariadenie zo servera.

**Zvolené riešenie** – prvá možnosť nepripadala do úvahy, pretože aplikácie s kvalitným a moderným používateľským rozhraním majú omnoho väčší úspech ako aplikácie s viac funkciami, ktoré nekladú príliš veľkú váhu na ovládanie. Preto je v aplikácii spojená druhá a tretia možnosť z dôvodu väčšej univerzálnosti. Vytváranie cesty pomocou webovej aplikácie je veľmi zaujímavé a môže byť implementované neskôr pri rozvíjaní aplikácie.

### Údaje cesty

Tento modul umožňuje používateľovi vytvárať alebo upravovať cestu. Z toho dôvodu musí obsahovať prvky, ktoré mu umožnia vykonávať zmeny údajov cesty. Modul bude umožňovať nasledujúce funkcie:

- **Zobraziť mapu s objektmi cesty** – mapa bude obsahovať objekty cesty, ktoré cesta obsahuje. Ak už niektoré objekty cesty alebo odpovede k ich otázkam budú mať so sebou asociované nasledujúce objekty cesty, tak tieto asociácie budú reprezentované čiarami medzi týmito objektami.
- **Zmeniť názov cesty** – umožní používateľovi zmeniť názov cesty.
- **Zmeniť obrázok cesty** – spustí Android Intent, z ktorého sa budú dať vybrať rôzne aplikácie, ktoré ponúkajú obrázky. Takýmito sú napríklad galéria, Google Drive, Dropbox atď. Po vybratí obrázku sa spustí okno, kde bude môcť používateľ orezať obrázok tak, aby najlepšie zobrazoval požadovaný detail. Pokiaľ používateľ nevyberie žiadny obrázok, tak bude použitý predvolený obrázok cesty.
- **Zmeniť odhadovaný čas cesty** – umožní používateľovi zadať číslo, ktoré reprezentuje odhadovaný čas cesty v minútach.
- **Zmeniť náročnosť cesty** – zobrazí používateľovi na výber z viacerých možností, ktoré budú reprezentovať náročnosť cesty.
- **Pridanie nového objektu cesty** – zobrazí používateľovi údaje o objekte cesty, ktoré bude môcť následne upravovať.
- **Zoznam objektov cesty** – obsahuje zoznam už vytvorených objektov cesty. Používateľ bude môcť objekty mazať alebo upravovať. Pokiaľ by chcel používateľ objekt upravovať, tak sa prepne obrazovka na jeho detail a zobrazí už vyplnené údaje, ktoré bude možné upravovať.
- **Uloženie cesty** – uloží cestu do lokálnej databázy a vráti sa späť do modulu prehľadu ciest.

## Údaje o objekte cesty

Cesta je tvorená aspoň dvomi objektmi cesty, ktoré si môžeme predstaviť ako záchytné body pre používateľa. Hlavnou časťou objektu cesty je jeho poloha, ktorá určuje kde máme objekt cesty zobraziť. Objekt cesty môže byť bodom alebo šípkou, kde bod predstavuje miesto, kam sa má používateľ dostaviť, avšak šípka predstavuje orientovanú šípku, ktorá zobrazuje smer, ktorým sa má používateľ vydať. Pri pridávaní nového objektu cesty bude modul poskytovať nasledujúce funkcie:

- **Mapa** – kliknutím na mapu sa pridá na mapu bod, ktorý vyplní údaje o jeho polohe do príslušných polí s týmito položkami.
- **Meno objektu** – pole, do ktorého sa bude dať zadať meno objektu.
- **Zemepisná šírka** – určuje zemepisnú šírku objektu cesty.
- **Zemepisná dĺžka** – určuje zemepisnú dĺžku objektu cesty.
- **Rádus** – počet metrov, ktoré určujú kruh okolo objektu cesty. Keď sa používateľ dostaví do tohto kruhu, tak mu tento objekt bude uznaný ako prejdený. Ak je objekt šípkou, tak je pre používateľa viditeľná iba pokiaľ sa nachádza v tomto kruhu.

- **Výber typu objektu** – jednoduchá možnosť vybrať či je objekt šípkou alebo bodom.
- **Orientácia šípky** – určuje uhol otočenia šípky.
- **Pridať otázku** – zviaže s objektom novú otázku. Pokiaľ je s objektom už otázka vytvorená, tak ju bude možné zmazať alebo upraviť.
- **Vybrať typ odpovede** – ak je s objektom zviazaná otázka, je možné vybrať, či používateľ bude mať na výber z viacerých možností alebo bude musieť napísať celú odpoveď. Ak bude používateľ musieť napísať celú odpoveď, tak môže byť s jednou odpoveďou spojených viac správnych možností, ktorých oddelovačom bude čiarka, pričom sa do úvahy neberie veľkosť písmen.
- **Vybratie nasledujúceho objektu** – ak s objektom nie je zviazaná otázka, tak aplikácia umožní používateľovi vybrať predvolený nasledujúci objekt. Ak je s objektom zviazaná otázka, tak každá odpoveď musí obsahovať nasledujúci objekt. Ak bude musieť používateľ napísať celú odpoveď, tak bude musieť byť vybraný aj jeden objekt navyše, ktorý bude reprezentovať nesprávnu odpoveď.
- **Pridať nápovedy** – pridá nápovedy k danému objektu a počet trestných minút, ktoré sa používateľovi za ňu pripočítajú. Taktiež bude možné zmeniť ich poradie.
- **Uložiť objekt cesty** – uloží objekt v danej ceste.

### 2.2.3 Prechádzanie cesty

#### Predsieň hry

Po prejení do tohto modulu zo zoznamu ciest dostaneme ako parameter cestu, ktorú chce používateľ prejsť. Zobrazíme rozšírené informácie o ceste, ktoré sa nezmestili do zoznamu ciest v module prehľadu ciest. Taktiež zobrazíme dve tlačidlá, ktorými si používateľ bude môcť vybrať typ hry:

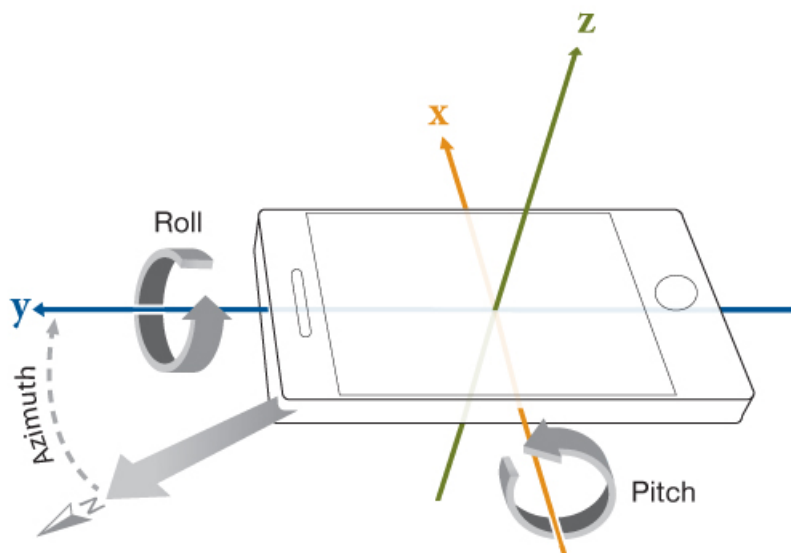
- **Tlačidlo pre hru jedného hráča** – prepne modul do režimu prechádzania cesty.
- **Tlačidlo pre hru viacerých hráčov** – zobrazí používateľovi priateľov, ktorým môže poslať pozvánky, ktoré im prídu ako notifikácie, alebo môže vyhľadať používateľov v blízkosti zariadenia. Po odoslaní pozvánok sa zobrazí obrazovka, kde sa zobrazia ich stav. Pred začatím hry nahrá zakladateľ hry viacerých hráčov cestu na server ako dočasnú cestu. Po spustení hry viacerých hráčov si musia všetci účastníci prechádzanú cestu stiahnuť zo servera. Potom sa všetci budú musieť dostaviť na štart cesty a hra sa môže začať.

## Hra

Väčšiu časť obrazovky zaberá pohľad kamery, ktorý sa automaticky zaostruje. Túto časť prekrýva ďalšia transparentná časť, do ktorej sa bude vykreslovať grafika v OpenGL [2]. Ak bude používateľ otočený smerom, v ktorom sa nachádza aktuálny objekt cesty, tak sa zobrazí v strede obrazovky. Ak bude otočený tak, že ho už kamera nezachytí, tak ho používateľ nebude vidieť.

## Získanie natočenia zariadenia

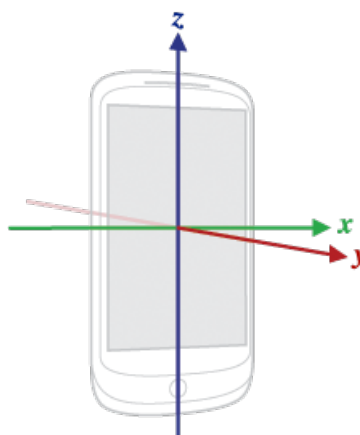
Na získanie natočenia zariadenia použijeme Rotation Vector senzor, ktorý je prístupný na zariadeniach s operačným systémom Android. Pomocou tohto senzora získame hodnoty azimuth, pitch a roll, ktoré si môžeme predstaviť podľa obrázku 2.5:



Obr. 2.5: Zobrazenie hodnôt azimuth, pitch a roll, prevzaté z [3]

- **Azimuth** – rotácia okolo osi  $z$ . Je uhol medzi pozitívnou osou  $y$  a magnetickým severom.
- **Pitch** – rotácia okolo osi  $x$ . Pozitívny pitch začína, keď zariadenie leží rovno na stole, a pozitívna os  $z$  sa začne nakláňať smerom k pozitívnej osi  $y$ .
- **Roll** – rotácia okolo osi  $y$ . Pozitívny roll začína, keď zariadenie leží rovno na stole a pozitívna os  $z$  sa začne nakláňať smerom k pozitívnej osi  $x$ .

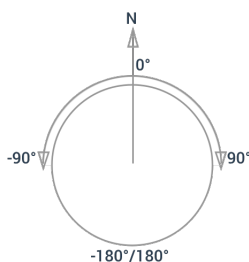
My natočenie získavame pre správne zobrazovanie rozšírenej reality, teda os  $y$  musí byť súbežná s osou kamery. Preto je nutné premapovať os  $z$  na os  $y$ . Výsledok je možné vidieť na obrázku 2.6. Tým získame správne hodnoty azimuth, pitch a roll, ktoré môžeme využiť v aplikácii.



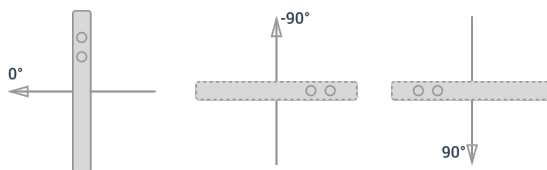
Obr. 2.6: Nový koordinačný systém zariadenia, prevzaté z [4]

### Získanie hodnôt vektora pohľadu kamery

Máme k dispozícii hodnoty azimut, pitch a roll a chceme vypočítať vektor pohľadu kamery v našom virtuálnom 3D svete. Pre naše potreby je dôležitý azimut a pitch, ktorých hodnoty sú zobrazené na obrázkoch 2.7 a 2.8



Obr. 2.7: Hodnoty azimutu voči severu

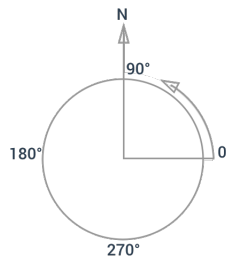


Obr. 2.8: Hodnoty pitch

Hodnoty uhla azimut premapujeme na hodnoty, s ktorými sa bežne pracuje v matematike, ktoré sú zobrazené na obrázku 2.9.

Hodnoty uhla pitch iba znegujeme, aby nám sínus z jeho uhla vracal správne hodnoty. Pitch a azimut nám teraz predstavujú sférické súradnice. Aby sme získali správne hodnoty vektora pohľadu kamery, musíme vykonať nasledujúce výpočty:

$$\begin{aligned} x &= \cos(\text{azimut}) \cdot \cos(\text{pitch}) \\ y &= \sin(\text{pitch}) \\ z &= \sin(\text{azimut}) \cdot \cos(\text{pitch}) \end{aligned} \tag{2.1}$$



Obr. 2.9: Obyčajné hodnoty uhla

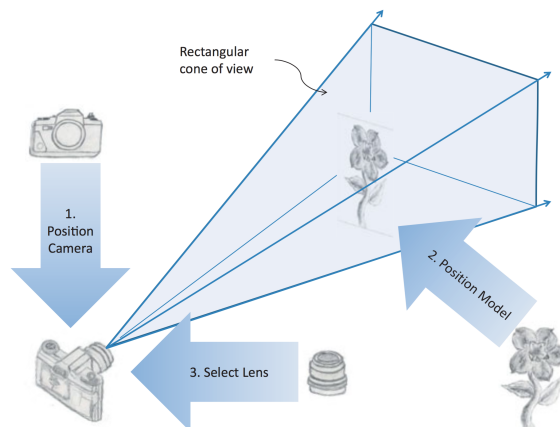
## Zobrazovanie objektov cesty

Vykresľovanie objektov cesty bude závisieť na ich type:

- **Bod** – bod budeme reprezentovať ako 2D obdĺžnik, ktorý bude vždy natočený smerom k zariadeniu. Na tomto 2D obdĺžniku budeme vykresľovať textúru.
- **Šípka** – šípka je reprezentovaná ako 3D objekt farbou, ktorú používateľ špecifikoval v nastaveniach aplikácie. Natočenie šípky vypočítame pomocou jej uhla a vykreslíme ju. Avšak ak by bol celý objekt rovnakej farby, tak by používateľ nevedel určiť jeho smer, z tohto dôvodu je nutné implementovať osvetlenie objektu. Preto si musíme pre každý vrchol šípky pamätať aj jeho normálový vektor.

Vykresľovanie pomocou OpenGL si môžeme predstaviť pomocou jednoduchšej analógie s kamerou, ktorá je vidieť na obrázku 2.10 a popisuje nasledovné transformácie:

1. **View transformácie** – posunieme kameru na miesto, odkiaľ chceme natáčať a nasmerujeme ju správnym smerom.
2. **Model transformácie** – posunieme objekty na správne miesto.
3. **Projection transformácie** – vyberieme objektív alebo nastavíme priblíženie.



Obr. 2.10: Príklad vykresľovania pomocou OpenGL, prevzaté z [5]

Každá transformácia je vyjadrená pomocou  $4 \times 4$  matice. Všetky transformácie vynásobíme v správnom poradí a finálnou maticou prenásobíme polohu nášho vrchola, čím dostaneme jeho správne súradnice vo virtuálnom svete.

Pomocou hodnôt natočenia môžeme vypočítať view transformáciu. Poloha nášho zariadenia bude vždy na počiatkových súradniciach  $[0, 0, 0]$  nášho virtuálneho 3D sveta. Súradnice našich objektov cesty v 3D budeme aproximovať podľa ich vzdialenosti v skutočnom svete a uhla, ktorý reprezentuje najkratšiu cestu k tomuto objektu. Čím ďalej v skutočnom svete, tým menšie by sa objekty mali zdať. Ale toto zmenšovanie nemôžeme aplikovať donekonečna, pretože v OpenGL musíme mať špecifikovanú vzdialenú priemetňu, preto musíme tieto hodnoty aproximovať. Týmto získame model transformácie. Nakoniec získame projection maticu, ktorej parametrami sú šírka a výška zariadenia (aby sme zachovali pomer strán vykresľovaných objektov pre zariadenia s rôznymi obrazovkami) a horizontálne zorné pole.

## Informácie pre používateľa

Používateľovi sa v hornom paneli budú zobrazovať nasledujúce informácie:

- **Prejdená vzdialenosť** – prejdená vzdialenosť v metroch rátaná od začiatku cesty, t.j. od štartovného objektu cesty.
- **Čas cesty** – celkový čas cesty rátaný od začiatku cesty.
- **Trestný čas** – trestný čas, ktorý používateľ „získal“ aktuálnym priechodom cesty. Je to súčet všetkých trestných minút za vybrané odpovede k otázkam a použité nápovedy.

Na obrazovke sa bude nachádzať aj tlačidlo pre zobrazenie nápovedy. Po kliknutí na toto tlačidlo sa aplikácia používateľa spýta, či je ochotný zobrazíť danú nápovedu za daný počet trestných bodov. Ak používateľ túto možnosť potvrdí, tak mu aplikácia zobrazí nápovedu ako jednoduchú správu priamo na obrazovke.

V spodnej časti obrazovky bude zobrazená vzdialenosť používateľa od nasledujúceho objektu cesty, kde zobrazovaná vzdialenosť môže byť:

- **Bod** – priamo vzdialenosť používateľa od bodu, kam sa má používateľ dostať.
- **Šípka** – vzdialenosť používateľa od šípky, pokiaľ k nej používateľ ešte nedorazil alebo sa nachádza v jej rádiuse. Inak sa zobrazí vzdialenosť od nasledujúceho bodu.

## Hra jedného hráča

Príklad prechádzania cesty je popísaný v podkapitole 2.1.4. Po vstupe do hry sa hráčovi zobrazí štartovný objekt cesty. Keď používateľ k tomuto objektu dorazí, tak sa ho aplikácia spýta, či je pripravený začať hru. Ak áno, tak sa spustí časomiera a začne sa rátať dĺžka cesty. Bod bude predstavovať miesto, kam sa má používateľ dostať, a pre používateľa je to len jednoduché priamočiare nasledovanie. Šípka je v tomto iná a zobrazuje iba smer, ktorým by sa mal používateľ vydať. Pokiaľ používateľ prejde bodom, tak sa rovno zobrazí nasledujúci bod.

Pokiaľ používateľ prejde šípkou, tak sa stále táto šípka môže zobrazovať, ak sa používateľ bude nachádzať v jej rádiuse. Inak sa zobrazí nasledujúci bod. Zobrazenie šípky, pokiaľ sa používateľ nachádza v jej okolí, je prítomné z toho dôvodu, keby používateľ odbočil zlým smerom a chcel by sa vrátiť a prehodnotiť svoje rozhodnutie. Vhodné umiestnenie šípky je na križovatke, kde usmerní používateľa, ktorou cestou by sa mal vydať.

Pri dorazení k bodu cesty môže byť používateľovi položená otázka, ktorej odpovede môžu vetviť cestu rôznymi smermi. Otázky môžu byť dvoch typov:

- **Výber odpovede** – používateľovi sa zobrazí okno s možnosťami, z ktorých si používateľ musí vybrať. Odpovede budú zobrazené v náhodnom poradí.
- **Napísanie správnej odpovede do textového poľa** – používateľ bude musieť napísať správnu odpoveď. Nebude sa brať do úvahy veľkosť písmen. Pri tejto možnosti musí byť špecifikovaný jeden predvolený objekt cesty, ktorý predstavuje nesprávnu odpoveď.

Pokiaľ si používateľ nevyberie odpoveď, nebude môcť pokračovať ďalej. Viacero odpovedí môže reprezentovať ten istý nasledujúci objekt cesty.

Ak používateľ dorazí do stavu, z ktorého už nebude môcť pokračovať na ďalší objekt cesty, tak hra skončí.

## Hra viacerých hráčov

Princíp hry viacerých hráčov je rovnaký ako pri hre jedného hráča. Všetci hráči sa musia dostaviť na štartovný objekt cesty a potvrdiť, že sú pripravení začať hru. Hra začne vtedy, keď svoju pripravenosť potvrdia všetci hráči. Používatelia budú vidieť na obrazovke všetkých svojich protihráčov ako avatarov podobne, ako to je pri zobrazovaní bodu. Teda ichvidia iba vtedy, keď budú natočený rovnakým smerom, v ktorom sa nachádza ich protihráč. Vyhráva ten hráč, ktorý bude mať po pričítaní trestných bodov najmenší celkový čas. Po ukončení cesty sa používateľovi zobrazí obrazovka aktuálnych časov ostatných hráčov. Používateľ môže počkať na ich dokončenie alebo z hry odísť. Zobrazenie polohy ostatných hráčov nie je pre hru kompromitujúce, pretože používatelia budú v rôznych častiach cesty z dôvodu vetvenia, tým pádom im poloha protihráča neprezradí nič, čo by mohli využiť k svojmu zisku.

## 2.3 Server

Server si takisto musí o ceste pamätať niektoré údaje navyše od tých, ktoré boli popísané v podkapitole 2.1. Týmito údajmi sú:

- **Id autora** – musíme identifikovať používateľa, ktorý má právo aktualizovať alebo meniť danú cestu.
- **Informácia, či je cesta dočasná** – je to nutné kvôli tomu, aby sme vedeli, ktoré cesty boli nahrané na server iba z dôvodu hry viacerých hráčov a môžu byť po určitom čase zmazané.
- **Čas vytvorenia** – je nutné si pamätať čas vytvorenia cesty na serveri, aby sme náhodou nezmazali dočasnú cestu, ktorú používateľ práve nahral pre potreby hry viacerých hráčov.



### 2.3.1 Implementácia servera

Server by mal byť hlavne stabilný a efektívny, preto je veľmi dôležitá jeho implementácia. Existuje veľmi veľa jazykov a frameworkov, pomocou ktorých je veľmi jednoduché a priamočiare napísať jednoduchý server. Medzi takéto jazyky patrí aj Java, ktorej podmnožinu sme využívali aj pri písaní mobilnej aplikácie, preto je jej použitie vhodné a praktické. Máme na výber viacero možností, z ktorých zostali na základe dôvodov, medzi ktorými boli hlavne praktickosť a elegantnosť kódu, nasledovné možnosti:

- **Vlastná implementácia** – práca a kód by neboli veľmi jednoduché, pretože by sme museli ošetrovať veľké množstvo výnimiek a písať väčšie množstvo zdĺhavého kódu.
- **Spring Boot [6]** – Spring Boot je zjednodušená vrstva postavená nad frameworkom Spring, ktorá zabaľuje všetky jeho významné paradigmy a zameriava sa na jednoduchý vývoj a jeho nasadenie.
- **Play framework [7]** – je programátorsky priateľský framework, ktorý dokáže zobrazíť zmeny v priebehu sekúnd, čo zvyšuje produktivitu. Taktiež poskytuje neblokujúce I/O<sup>7</sup> operácie a lepšie zobrazovanie chýb.

**Zvolené riešenie** – pre implementáciu servera bol nakoniec zvolený Spring Boot najmä kvôli skúsenostiam riešiteľa s platformou Spring.

### 2.3.2 Reprezentácia dát pri posielaní cez sieť

Je vždy nutné sa dohodnúť na správnom protokole, aby si dva komunikujúce uzly navzájom rozumeli. Dáta nie je nutné šifrovať, pretože neobsahujú žiadne citlivé údaje z toho dôvodu, lebo prihlasovanie je implementované pomocou Google účtu, čím nám stačí poselať iba reťazec predstavujúci token používateľa, ktorý na serveri kontrolujeme. Možnými riešeniami reprezentácie dát boli:

- **Vlastný formát** – toto riešenie je vhodné pre veľmi jednoduché štruktúry, ale v našom prípade by nebolo príliš výhodné, pretože štruktúra cesty môže byť zložitá a podobné technológie sú už vyvinuté a dobre implementované.
- **Reprezentácia pomocou XML<sup>8</sup>** – XML je veľmi rozšírené a má veľmi solídnu podporu. Jeho hlavnou nevýhodou je, že má väčšiu veľkosť ako rovnaké dáta zapísané vo formáte JSON.
- **Reprezentácia pomocou JSON** – hlavnou výhodou formátu JSON je to, že existuje mnoho serializačných knižníc, ktoré vedia previesť Java objekt na JSON objekt a naopak. Príkladmi takýchto knižníc sú napríklad Google Gson [8] alebo Jackson [9]. Takisto existuje priama podpora JSON konverzie v Spring frameworku, ktorý je použitý na serveri.

**Zvolené riešenie** – bola zvolená reprezentácia pomocou JSON formátu, najmä kvôli znamenitej podpore na oboch stranách komunikácie.

---

<sup>7</sup>Input/Output

<sup>8</sup>eXtensible Markup Language, <https://en.wikipedia.org/wiki/XML>

### 2.3.3 Implementácia webových služieb

Na výber boli iba 2 možnosti:

- **SOAP<sup>9</sup> web services** – SOAP web services je protokol, ktorý mimo iné umožňuje volať metódy na vzdialenom serveri. Je potrebné vytvárať zdĺhavé WSDL<sup>10</sup> súbory, ktoré popisujú serverovú časť. Tento protokol je veľmi univerzálny a preto aj veľmi nepraktický, pretože je náročné s ním pracovať.
- **RESTful API** – REST funguje na princípe používania GET, PUT, POST a DELETE metód volaní protokolu HTTP<sup>11</sup>. Takýmto prístupom sa môžu zdieľať zdroje napr. pomocou princípu CRUD<sup>12</sup>. Hlavnou výhodou je, že implementácia RESTful API je priamočiara a veľmi jednoduchá.

**Zvolené riešenie** – nakoniec sa lepším riešením stalo RESTful API. Najmä vďaka knižnici Retrofit [10] a frameworku Spring, ktoré umožňujú prácu s danou technológiou a jej jednoduchému a prívetivému konceptu.

### 2.3.4 Perzistencia dát

#### Databáza

V prvom rade je potrebné vybrať vhodnú databázu. Máme na výber medzi vstavanými databázami<sup>13</sup> priamo v projekte a obvyčajnými databázami, ktoré bežia nezávisle na aplikácii. Koncept vstavaných databáz je veľmi nápomocný vo fáze vývoja, pretože sú malé, rýchle, majú rýchly čas načítania a jednoduchú konfiguráciu, čím sa môžeme viac sústrediť na vývoj. Medzi najznámejšie vstavané databázy patria:

- HSQLDB [11]
- H2 [12]
- Derby [13]

Na základe údajov tabuľky 2.1 vyzerá, že H2 je rýchlejšia oproti jej konkurentom, preto bola zvolená ako databáza servera. Keďže výsledky tohto porovnania boli získané zo stránky H2 databázy, je možné, že boli použité testy, ktoré vyhovovali priamo H2 databáze.

#### Mapovanie objektov do databázy

Využijeme JPA<sup>14</sup>, čo je rozhranie popisujúce manažment relačných dát v Java aplikáciách. Ako implementáciu tohto rozhrania použijeme Hibernate [15], čo

<sup>9</sup>Simple Object Access Protocol, <https://en.wikipedia.org/wiki/SOAP>

<sup>10</sup>Web Services Description Language, [https://en.wikipedia.org/wiki/Web\\_Services\\_Description\\_Language](https://en.wikipedia.org/wiki/Web_Services_Description_Language)

<sup>11</sup>Hypertext Transfer Protocol, [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

<sup>12</sup>Create, read, update and delete, [https://en.wikipedia.org/wiki/Create,\\_read,\\_update\\_and\\_delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete)

<sup>13</sup>embedded database, [https://en.wikipedia.org/wiki/Embedded\\_database](https://en.wikipedia.org/wiki/Embedded_database)

<sup>14</sup>Java Persistence API, [https://en.wikipedia.org/wiki/Java\\_Persistence\\_API](https://en.wikipedia.org/wiki/Java_Persistence_API)

| Test Case                   | Unit | H2      | HSQLDB  | Derby   |
|-----------------------------|------|---------|---------|---------|
| Simple: Init                | ms   | 1019    | 1907    | 8280    |
| Simple: Query (random)      | ms   | 1304    | 873     | 1912    |
| Simple: Query (sequential)  | ms   | 835     | 1839    | 5415    |
| Simple: Update (sequential) | ms   | 961     | 2333    | 21759   |
| Simple: Delete (sequential) | ms   | 950     | 1922    | 32016   |
| Simple: Memory Usage        | MB   | 21      | 10      | 8       |
| BenchA: Init                | ms   | 919     | 2133    | 7528    |
| BenchA: Transactions        | ms   | 1219    | 2297    | 8541    |
| BenchA: Memory Usage        | MB   | 12      | 15      | 7       |
| BenchB: Init                | ms   | 905     | 1993    | 8049    |
| BenchB: Transactions        | ms   | 1091    | 583     | 1165    |
| BenchB: Memory Usage        | MB   | 17      | 11      | 8       |
| BenchC: Init                | ms   | 2491    | 4003    | 8064    |
| BenchC: Transactions        | ms   | 1979    | 803     | 2840    |
| BenchC: Memory Usage        | MB   | 19      | 22      | 9       |
| Executed statements         | #    | 1930995 | 1930995 | 1930995 |
| Total time                  | ms   | 13673   | 20686   | 105569  |
| Statements per second       | #    | 141226  | 93347   | 18291   |

Tabuľka 2.1: Tabuľka porovnania najznámejších vstavaných databáz podľa [14]

je ORM<sup>15</sup> framework. V našom projekte iba vytvoríme modelové triedy našich tabuliek a popíšeme vzájomné vzťahy medzi nimi. Pre prácu s databázou budeme využívať Spring Data JPA, čo nám poskytuje príjemné prostredie.

---

<sup>15</sup>Object/Relational mapping, [https://en.wikipedia.org/wiki/Object-relational\\_mapping](https://en.wikipedia.org/wiki/Object-relational_mapping)

## 3. Uživatelská dokumentácia

Táto kapitola popisuje a vysvetľuje prácu s aplikáciou. Z väčšej časti obsahuje obrázky jednotlivých obrazoviek a objasňuje význam ich jednotlivých komponentov. Obrázky obrazoviek sú zábermi obrazovky reálneho zariadenia LG P880 bežiacom na verzii Android KitKat 4.4.4. Jednotlivé komponenty budú obsahovať anglický text.

### 3.1 Požiadavky na zariadenie

Pre spustenie aplikácie je nutné mať zariadenie so systémom Android s verziou aspoň 4.1 Jelly Bean. Toto zariadenie musí disponovať gyroskopom, akcelerometrom, magnetickým kompasom a kamerou. Pre najlepšie výsledky by malo taktiež obsahovať GPS modul, aj keď pre funkčnosť to nie je nutné, stačí možnosť nájsť polohu pomocou GSM alebo Wi-Fi sietí.

### 3.2 Inštalácia

Inštalácia by mala byť vykonaná cez obchod Google Play, ktorý by sa mal nachádzať na každom certifikovanom zariadení.

### 3.3 Prehľad ciest

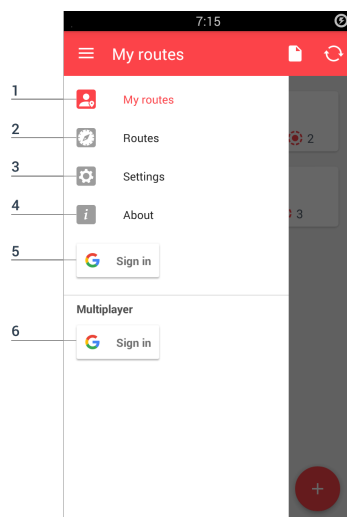
Prehľad ciest sa skladá z nasledovných obrazoviek:

- **Zoznam lokálne uložených ciest** – obrazovka, ktorej hlavným prvkom je zoznam lokálne uložených ciest. Tento zoznam je podrobnejšie popísaný v sekcii 3.3.2.
- **Zoznam vzdialene uložených ciest** – obrazovka, ktorá obsahuje zoznam, ktorý sa aktualizuje jeho posúvaním, pretože sťahovať všetky cesty zo servera naraz by bolo neefektívne.
- **Nastavenia** – obrazovka, v ktorej môže používateľ zmeniť niektoré nastavenia aplikácie.
- **Informácie o aplikácii** – obrazovka, ktorá obsahuje názov, verziu, autora a dôvod vzniku aplikácie.

Medzi týmito obrazovkami sa jednoducho prepína pomocou vysúvacieho menu popísaného v 3.3.1.

#### 3.3.1 Vysúvacie menu

Základné vysúvacie menu, keď užívateľ nie je prihlásený pomocou účtov, vyzerá podľa obrázku 3.1.

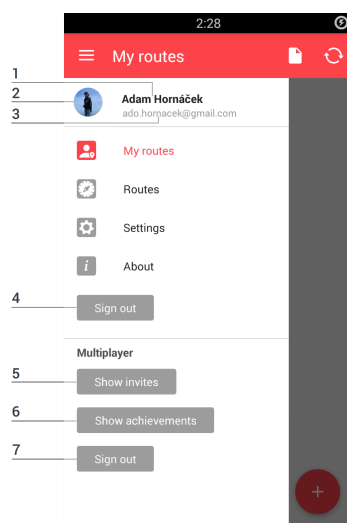


Obr. 3.1: Vysúvacie menu

Popis jednotlivých komponentov na obrázku 3.1:

1. Položka, ktorá reprezentuje zoznam lokálne uložených ciest.
2. Položka, ktorá reprezentuje zoznam vzdialene uložených ciest.
3. Položka, ktorá reprezentuje nastavenia aplikácie.
4. Položka, ktorá zobrazí základné informácie o aplikácii.
5. Tlačidlo, ktorým sa prihlasuje pomocou Google účtu.
6. Tlačidlo, ktorým sa prihlasuje do Google Play Games.

Pokiaľ je používateľ prihlásený do oboch účtov, budú sa mu zobrazovať možnosti a informácie zobrazené na obrázku 3.2.



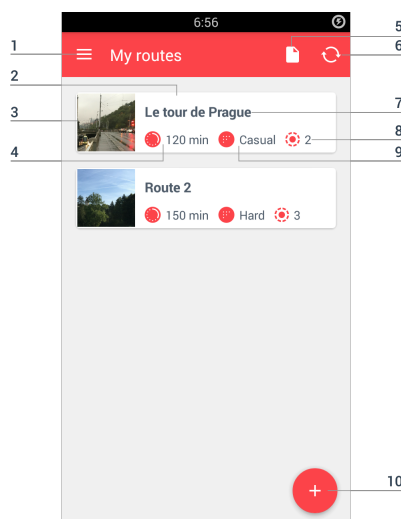
Obr. 3.2: Vysúvacie menu, pokiaľ je používateľ prihlásený pomocou účtov

Popis jednotlivých komponentov na obrázku 3.2:

1. Meno prihláseného používateľa pomocou Google účtu.
2. Obrázok prihláseného používateľa pomocou Google účtu.
3. Email prihláseného používateľa pomocou Google účtu.
4. Tlačidlo, ktorým sa používateľ odhlási z Google účtu.
5. Tlačidlo, ktorým sa zobrazia pozvánky od ostatných používateľov.
6. Tlačidlo, ktorým sa zobrazí obrazovka bonusov.
7. Tlačidlo, ktorým sa používateľ odhlási z Google Play Games.

### 3.3.2 Zoznam lokálne uložených ciest

Obrazovka vyzerá podľa obrázku 3.3.



Obr. 3.3: Zoznam lokálne uložených ciest

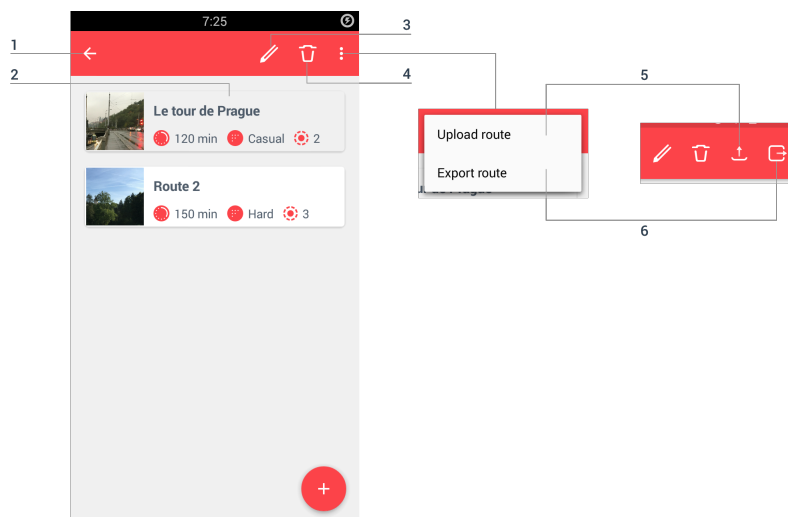
Popis jednotlivých komponentov na obrázku 3.3:

1. Tlačidlo, ktorým sa vysunie vysúvacie menu.
2. Položka, ktorá reprezentuje jednu lokálne uloženú cestu.
3. Obrázok cesty.
4. Odhadovaný čas cesty.
5. Tlačidlo, ktorým sa zobrazí obrazovka načítania cesty zo súboru.
6. Tlačidlo, ktorým sa spustí synchronizácia ciest so serverom.
7. Názov cesty.
8. Počet objektov cesty, ktoré cesta obsahuje.

9. Náročnosť cesty.

10. Tlačidlo, ktorým začneme vytvárať novú cestu.

Pokiaľ prst dlhšie podržíme na jednej položke, tak sa táto položka stane aktívnou a zobrazia sa nám viaceré možnosti, ktoré vyzerajú podľa obrázku 3.4. Na zariadeniach s menšou obrazovkou sa všetky možnosti nezmestia na obrazovku, preto obrázok obsahuje aj obrázok menu na väčších zariadeniach.



Obr. 3.4: Obrázok aktívnej položky zoznamu lokálne uložených ciest

Popis jednotlivých komponentov na obrázku 3.4:

1. Tlačidlo, ktorým sa zruší výber položky.
2. Aktívna položka.
3. Tlačidlo, ktorým sa spustí úprava cesty.
4. Tlačidlo, ktorým sa cesta zmaže. Pokiaľ je cesta uložená na serveri, tak sa pokúsi o jej zmazanie aj na ňom.
5. Tlačidlo, ktorým sa cesta nahrá na server.
6. Tlačidlo, ktorým sa spustí ukladanie cesty do súboru.

## 3.4 Vytváranie cesty

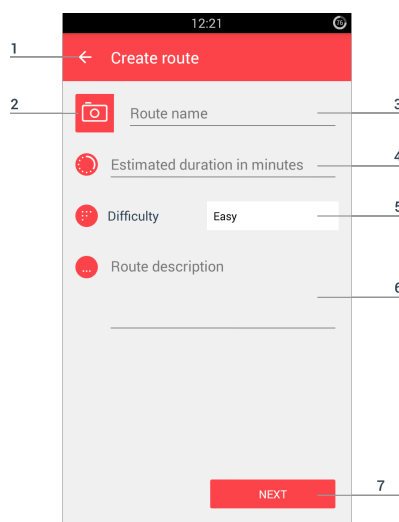
Úprava cesty je takmer identická s postupom pri vytváraní cesty, ale všetky polia už sú predvyplnené položkami upravovanej cesty. Modul vytvárania cesty obsahuje nasledovné obrazovky:

- **Základné informácie o ceste** – obrazovka obsahuje polia pre vyplnenie základných informácií o ceste. Z nej sa dá presunúť do zoznamu vytvorených objektov cesty. Viac informácií v sekcii 3.4.1.

- **Zoznam vytvorených objektov cesty** – zobrazuje všetky už vytvorené objekty cesty. Obsahuje tlačidlo, ktorým sa prejde na obrazovku vytvárania objektu cesty, a tlačidlo, ktorým sa prejde na obrazovku zobrazenia vytvorených bodov. Viac informácií v sekcii 3.4.2.
- **Vytváranie objektu cesty** – obrazovka, ktorá obsahuje polia pre vyplnenie potrebných informácií o objekte cesty. Viac informácií v sekcii 3.4.3.
- **Pridávanie otázky** – obrazovka, ktorá obsahuje potrebné polia pre vytvorenie otázky objektu cesty. Viac informácií v sekcii 3.4.4.
- **Vyberanie nápovied** – obrazovka, ktorá obsahuje zoznam nápovied, do ktorého sa dajú pridávať nové nápovedy a meniť poradie už vytvorených nápovied. Viac informácií v sekcii 3.4.5.
- **Výber polohy objektu cesty** – obrazovka, ktorá obsahuje mapu, na ktorú sa dá klikať, čím vyberieme súradnice polohy objektu cesty. Viac informácií v sekcii 3.4.6.
- **Zobrazenie objektov cesty na mape** – obrazovka, ktorej hlavným prvkom je mapa, ktorá obsahuje všetky už vytvorené objekty cesty. Užívateľsky príjemným spôsobom umožní objekty cesty spájať, čím sa bude vytvárať priebeh cesty. Viac informácií v sekcii 3.4.7.

### 3.4.1 Základné informácie o ceste

Obrazovka vyzerá podľa obrázku 3.5.



Obr. 3.5: Obrazovka základných informácií o ceste

Popis jednotlivých komponentov na obrázku 3.5:

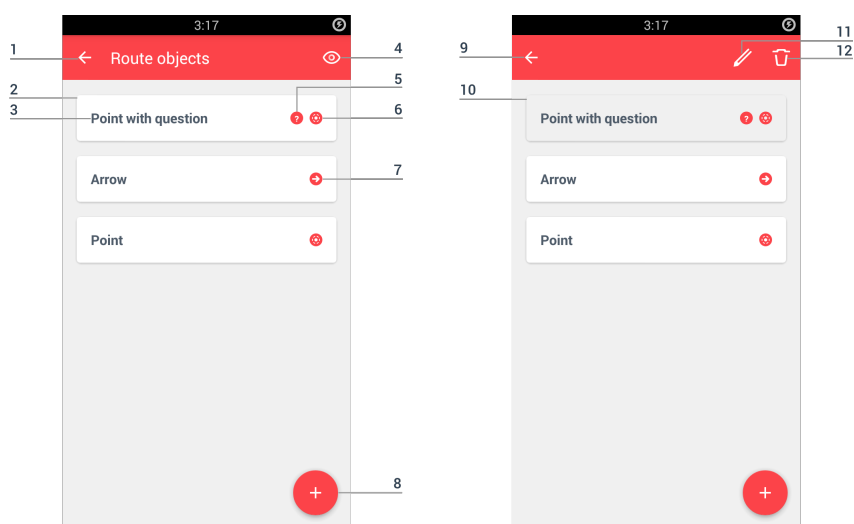
1. Tlačidlo vrátenia sa do modulu prehľadu ciest. Po stlačení sa používateľovi zobrazí dialógové okno, ktoré sa ho spýta, či naozaj chce opustiť modul vytvárania cesty. Pokiaľ túto možnosť potvrdí, tak sa aplikácia prepne do modulu prehľadu ciest a nevykoná žiadne zmeny.



2. Náhľad obrázku cesty. Pokiaľ používateľ na tento obrázok klikne, tak sa mu zobrazí aktivita výberu obrázku a jeho následného orezania, po ktorom sa zobrazí aktuálny náhľad obrázku.
3. Pole pre vyplnenie názvu cesty.
4. Pole pre vyplnenie odhadovaného času prejdenia cesty.
5. Výber obtiažnosti cesty.
6. Pole pre dlhší popis cesty.
7. Tlačidlo, ktorým sa prepne na obrazovku zoznamu objektov cesty.

### 3.4.2 Zoznam vytvorených objektov cesty

Obrazovka má vzhľad podľa obrázku 3.6.



Obr. 3.6: Obrazovka zoznamu vytvorených objektov cesty

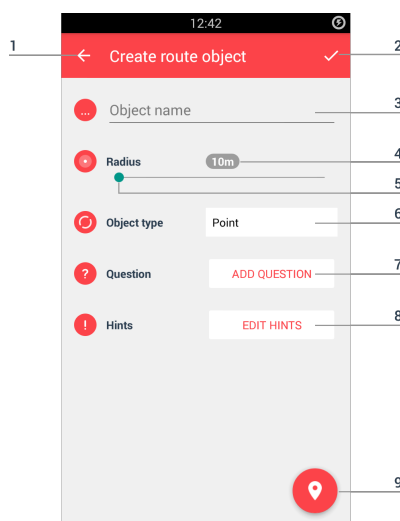
Popis jednotlivých komponentov na obrázku 3.4.2:

1. Tlačidlo, ktorým sa používateľ prepne na obrazovku základných informácií o ceste.
2. Položka zoznamu, ktorá reprezentuje jeden objekt cesty. Pokiaľ sa na ňu klikne, tak sa stane aktívnou.
3. Názov objektu cesty.
4. Tlačidlo, ktorým sa prepne na obrazovku zobrazenia objektov cesty na mape.
5. Položka, ktorá oznamuje, že s daným objektom cesty je zviazaná otázka.
6. Položka, ktorá oznamuje, že daný objekt cesty je bodom.
7. Položka, ktorá oznamuje, že daný objekt cesty je šípkou.

8. Tlačidlo, ktorým sa prepne na obrazovku vytvárania objektu cesty.
9. Tlačidlo, ktorým sa zruší výber položky.
10. Aktívna položka.
11. Tlačidlo, ktoré predstavuje úpravu objektu cesty. Po jeho stlačení sa zobrazí obrazovka vytvárania objektu cesty s vyplnenými hodnotami objektu cesty aktívnej položky.
12. Tlačidlo, ktorým sa zmaže objekt cesty reprezentovaný aktívnou položkou.

### 3.4.3 Vytváranie objektu cesty

Obrazovka vyzerá podľa obrázku 3.7.



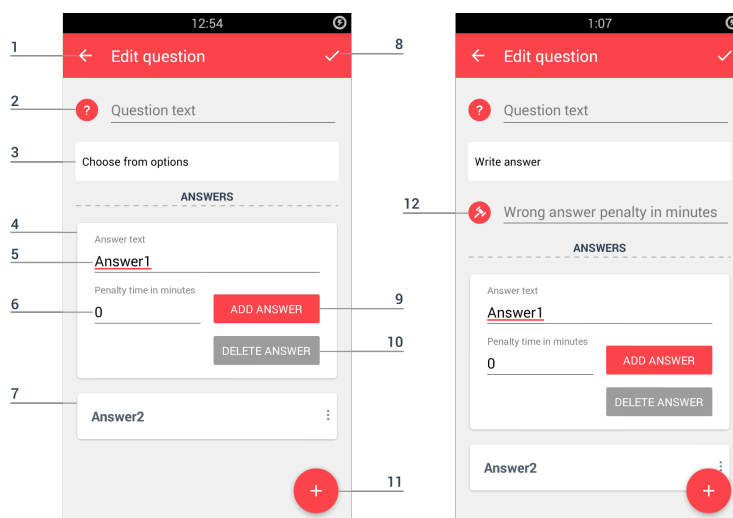
Obr. 3.7: Obrazovka vytvárania objektu cesty

Popis jednotlivých komponentov na obrázku 3.7:

1. Tlačidlo, ktorým sa používateľ vráti späť na zoznam objektov cesty. Žiadne vykonané zmeny sa neuložia.
2. Tlačidlo, ktorým sa vytvorí nový objekt, prípadne uložia zmeny objektu. Po uložení sa aplikácia prepne na obrazovku zoznamu vytvorených objektov cesty.
3. Pole, do ktorého sa vpíše meno objektu.
4. Text, ktorý oznamuje veľkosť rádiusu.
5. Posuvná položka, ktorá mení veľkosť rádiusu.
6. Výber typu objektu – bod alebo šípka.
7. Tlačidlo, ktoré spustí obrazovku výberu otázky.
8. Tlačidlo, ktoré spustí obrazovku nápoved.
9. Tlačidlo, ktorým sa spustí obrazovka výberu polohy objektu cesty.

### 3.4.4 Pridávanie otázky

Obrazovka pridávania otázky vyzerá ako na obrázku 3.8.



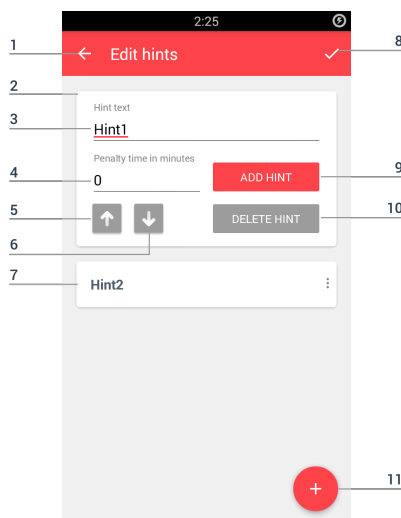
Obr. 3.8: Obrazovka vytvárania otázky

Popis jednotlivých komponentov na obrázku 3.8:

1. Tlačidlo, ktorým sa prepne na obrazovku vytvárania objektu cesty, pričom nebudú žiadne zmeny uložené.
2. Textové pole, kam je potrebné vyplniť text otázky.
3. Výber typu otázky – otázka s možnosťou výberu odpovede alebo vpisovaním odpovede.
4. Položka zoznamu odpovedí, ktorá reprezentuje aktívnu odpoveď.
5. Textové pole, kam sa vpisuje text odpovede.
6. Textové pole, kam sa vpisuje počet trestných minút pri výbere tejto odpovede.
7. Položka zoznamu odpovedí, ktorá reprezentuje neaktívnu odpoveď.
8. Tlačidlo, ktorým sa potvrdí vytváranie otázky. Po jeho stlačení sa zobrazí obrazovka vytvárania objektu cesty.
9. Tlačidlo, ktorým sa odznačí vybraná aktívna položka zoznamu odpovedí.
10. Tlačidlo, ktorým sa zmaže vybraná aktívna položka zoznamu odpovedí.
11. Tlačidlo, ktorým sa pridá nová odpoveď do zoznamu odpovedí.
12. Počet trestných minút nesprávnej odpovede pri otázke typu vpisovania odpovedi.

### 3.4.5 Vyberanie nápoved

Obrazovka vyberania nápoved vyzerá ako na obrázku 3.9.



Obr. 3.9: Obrazovka vyberania nápoved

Popis jednotlivých komponentov na obrázku 3.9:

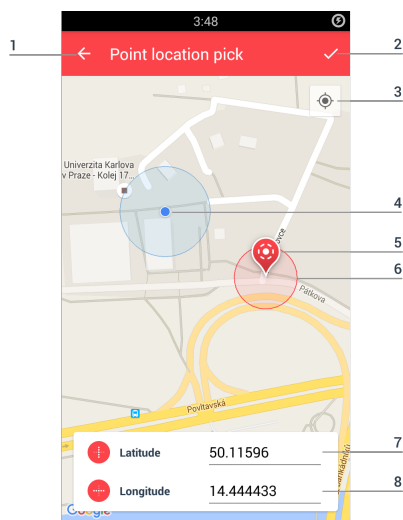
1. Tlačidlo, ktorým sa vrátíme späť na obrazovku vytvárania objektu cesty, vykonané zmeny nebudú uložené.
2. Aktívna položka zoznamu nápoved.
3. Textové pole, kam sa vpisuje text nápovedy.
4. Textové pole, kam sa vpisuje počet trestných minút.
5. Tlačidlo, ktorým sa poradie tejto nápovedy posunie smerom nahor.
6. Tlačidlo, ktorým sa poradie tejto nápovedy posunie smerom nadol.
7. Neaktívna položka zoznamu nápoved.
8. Uloží nápovedy a vráti sa na obrazovku vytvárania objektu cesty.
9. Tlačidlo, ktorým sa odznačí aktívna položka zoznamu nápoved.
10. Tlačidlo, ktoré zmaže túto nápovedu zo zoznamu nápoved.
11. Tlačidlo, ktoré pridá novú nápovedu do zoznamu nápoved.

### 3.4.6 Výber polohy objektu cesty

Rozlišujeme medzi výberom polohy podľa typu objektu, pretože pri výbere polohy pre typ šípka potrebujeme okrem polohy objektu určiť aj jej orientáciu.

## Bod

Obrazovka vyzerá tak, ako je zobrazená na obrázku 3.10.



Obr. 3.10: Obrazovka výberu polohy bodu

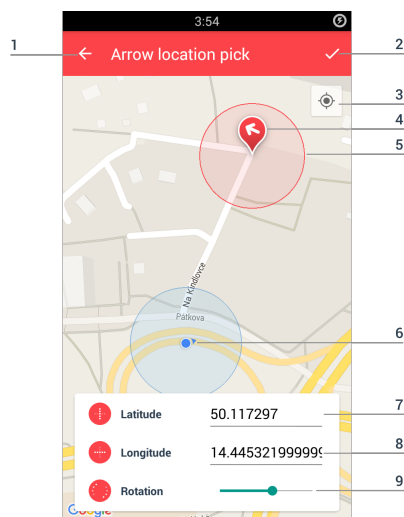
Popis jednotlivých komponentov na obrázku 3.10:

1. Tlačidlo, ktorým sa vrátíme späť na obrazovku vytvárania objektu. Vykonané zmeny nebudú uložené.
2. Tlačidlo, ktorým potvrdíme vykonané zmeny.
3. Tlačidlo, ktorým priblížime mapu na polohu zariadenia.
4. Poloha zariadenia.
5. Bod, ktorého pozíciu bude možné meniť podržaním prsta na ňom a jeho následným ťahaním.
6. Zobrazený rádius bodu.
7. Pole, kam sa dá vpísať zemepisná šírka bodu. Zmenením polohy bodu na mape sa takisto zmeny priemetnu do tohto poľa a zmenením tohto poľa sa premietnu zmeny na mape.
8. Pole, kam sa dá vpísať zemepisná dĺžka bodu. Takisto sa zmeny budú premietat na mapu a naopak.

## Šípka

Obrazovka vyzerá tak, ako je zobrazená na obrázku 3.11. Popis jednotlivých komponentov obrazovky:

- 1–3. Rovnaké významy ako pri vyberaní polohy bodu.
4. Šípka, ktorej pozíciu je možné meniť ťahaním. Orientácia bielej ikonky sa mení podľa orientácie šípky.



Obr. 3.11: Obrazovka výberu polohy a orientácie šípky

5. Rádus šípky.
6. Poloha zariadenia.
7. Pole, kam sa dá vpísať zemepisná šírka šípky. Zmeny sa premietajú na mape.
8. Pole, kam sa dá vpísať zemepisná dĺžka šípky. Zmeny sa premietajú na mape.
9. Posuvná položka, ktorá mení orientáciu šípky v rozsahu od 0 do 360 stupňov.

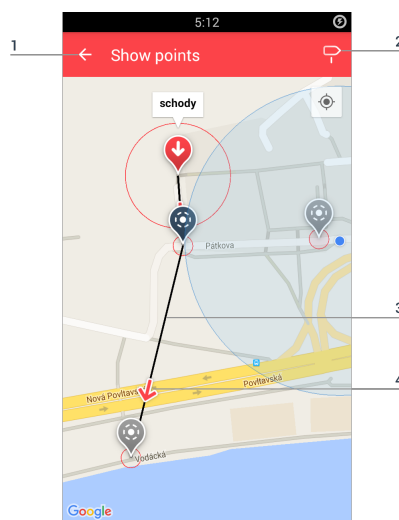
### 3.4.7 Zobrazenie objektov cesty na mape

Obrazovka obsahuje mapu a všetky objekty cesty, ktoré môžu mať rôzne farby na základe legendy, ktorá sa nachádza na obrázku 3.12.



Obr. 3.12: Legenda farieb objektov cesty

Príklad možného vzhľadu obrazovky je na obrázku 3.13.



Obr. 3.13: Obrazovka vyberania nasledujúcich bodov

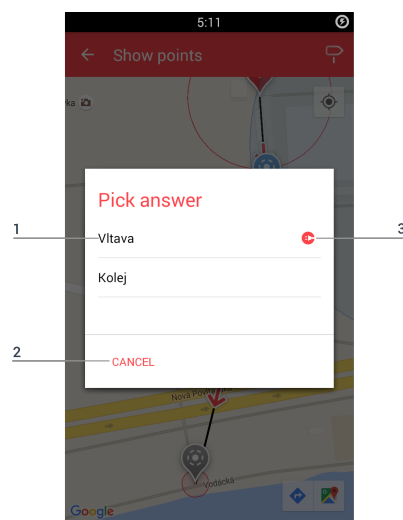
Popis jednotlivých komponentov na obrázku 3.13:

1. Tlačidlo, ktorým sa aplikácia prepne späť na obrazovku zoznamu objektov cesty.
2. Tlačidlo, ktorým sa začne výber štartovného objektu cesty.
3. Čiara, ktorá predstavuje spojenie (jeden objekt cesty je nasledujúcim pre druhý) medzi 2 objektmi cesty. Kliknutím na ňu sa toto spojenie zruší.
4. Objekt mapy, ktorý zobrazuje smer spojenia medzi dvoma objektmi. Po kliknutí naň sa toto spojenie zruší.

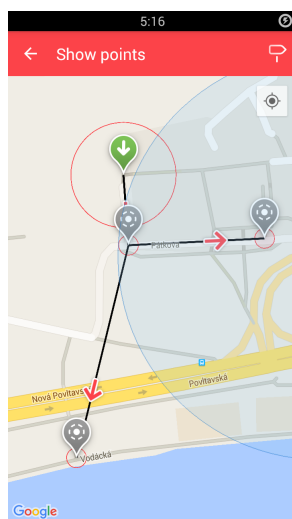
Po kliknutí na objekt cesty sa tento objekt stane aktívnym. Ak sa klikne na iný objekt, tak sa tento objekt nastaví ako nasledujúci pre aktívny objekt cesty. Ak je s aktívnym bodom spojená otázka, tak sa používateľovi zobrazí okno s odpoveďami otázky ako na obrázku 3.14. Po kliknutí na odpoveď sa pre túto odpoveď nastaví tento objekt ako nasledujúci. Kliknutím na aktívny objekt prestane byť tento objekt aktívny a je možné vybrať nový aktívny objekt. Týmto spôsobom sa musí vytvoriť celá cesta. Pokiaľ je používateľ so svojím výtvorom spokojný, tak môže kliknúť na tlačidlo výberu štartovného objektu cesty, ktoré je zobrazené na obrázku 3.13. Po kliknutí na toto tlačidlo sa cesta skontroluje a pokiaľ nenarušuje žiadne pravidlá hry, tak sa zobrazí obrazovka výberu štartovného objektu cesty, ktorá vyzerá ako na obrázku 3.15.

Popis jednotlivých komponentov na obrázku 3.14:

1. Položka, ktorá reprezentuje odpoveď.
2. Tlačidlo, ktorým sa zruší výber odpovedi.
3. Znázorňuje, že táto otázka už má so sebou zviazaný nasledujúci objekt cesty.



Obr. 3.14: Obrazovka výberu nasledujúceho objektu pre bod, s ktorým je zviazaná otázka



Obr. 3.15: Obrazovka výberu štartovného objektu cesty



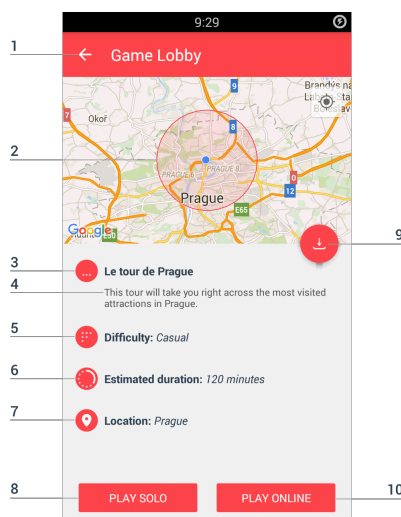
## 3.5 Prechádzanie cesty

Prechádzanie cesty sa skladá z troch základných obrazoviek:

- **Obrazovka detailu cesty** – obrazovka, ktorá obsahuje základné informácie o ceste, na základe ktorých sa používateľ môže rozhodnúť, či danú cestu chce stiahnuť do zariadenia alebo prechádzať. Viac informácií v sekcii 3.5.1.
- **Obrazovka hry** – obrazovka, ktorá obsahuje základné informácie o priebehu cesty a pomocou rozšírenej reality zobrazuje objekty cesty. Viac informácií v sekcii 3.5.2.
- **Obrazovka konca hry** – obrazovka, ktorá obsahuje rekapituláciu priebehu cesty. Viac informácií v sekcii 3.5.3.

### 3.5.1 Detail cesty

Obrazovka vyzerá podľa obrázku 3.16.



Obr. 3.16: Obrazovka detailu cesty

Popis jednotlivých komponentov na obrázku 3.16:

1. Tlačidlo, ktorým sa vrátíme späť do modulu prehľadu ciest.
2. Okruh, ktorý približne určuje polohu cesty. Je to najmenší okruh, ktorého stred je niektorý objekt cesty a ktorý obsahuje všetky objekty cesty.
3. Názov cesty.
4. Popis cesty.
5. Náročnosť cesty.
6. Odhadovaný čas.
7. Poloha štartovného objektu cesty. Najprv sa aplikácia pokúsi získať lokalitu, ak nie je dostupná, tak sa pokúsi získať administračnú oblasť a ak ani tá nie je dostupná, tak sa pokúsi získať názov krajiny.

8. Tlačidlo, ktorým sa spustí hra jedného hráča.
9. Tlačidlo, ktorým sa cesta uloží do zariadenia.
10. Tlačidlo, ktorým sa spustí hra viacerých hráčov.

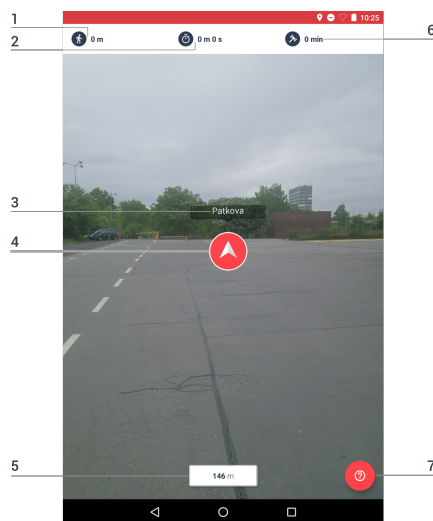
### 3.5.2 Hra

Táto podkapitola obsahuje obrázky obrazovky reálneho zariadenia Asus Nexus 7(2013) bežiacom na verzii Android 6.0.1. Hra môže pomocou rozšírenej reality zobrazovať 3 rôzne objekty:

- bod
- šípka
- hráč

#### Bod

Ak je aktuálnym objektom cesty bod alebo šípka, ktorú sme už navštívili a používateľ sa nenachádza v jej rádiuse, alebo šípka, ktorú sme ešte nenavštívili, tak obrazovka bude vyzeráť ako na obrázku 3.17.



Obr. 3.17: Obrazovka hry, ktorá zobrazuje bod

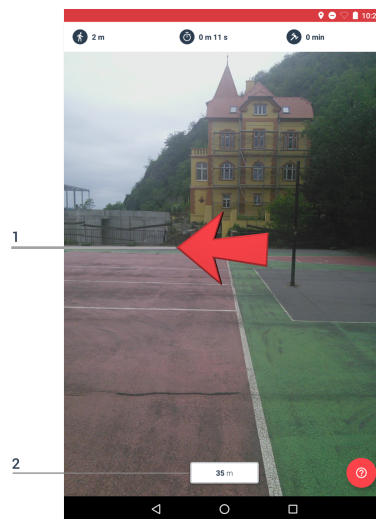
Popis jednotlivých komponentov na obrázku 3.17:

1. Aktuálne prejdená vzdialenosť, ktorá sa začína rátať až po dosiahnutí štartovného objektu cesty.
2. Aktuálny čas, ktorý sa začína rátať až po dosiahnutí štartovného objektu cesty.
3. Názov bodu, kam sa máme dostaviť. Pokiaľ názov bodu obsahuje znaky, ktoré nepatria do ASCII znakovkej sady, tak sa aplikácia pokúsi o odstránenie diakritiky a ak sa tam bude nachádzať znak, ktorý sa nebude dať zobraziť, tak ho aplikácia nahradí znakom, ktorý reprezentuje neznámy znak.

4. Textúra, ktorá reprezentuje bod.
5. Vzdialenosť od bodu.
6. Súčet trestných minút, ktoré používateľ doposiaľ „získal“.
7. Tlačidlo, ktorým sa používateľ spýta na nápovedu.

## Šípka

Pokiaľ je aktuálnym objektom cesty šípka a používateľ sa nachádza v jej rádiuse, tak obrazovka bude vyzeráť ako na obrázku 3.18.



Obr. 3.18: Obrazovka hry, ktorá zobrazuje šípku

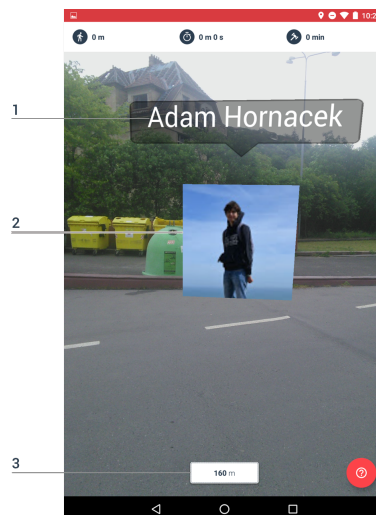
Popis jednotlivých komponentov na obrázku 3.18:

1. Šípka, ktorá je otočená smerom, ktorým by sa mal používateľ vydať.
2. Vzdialenosť od šípky.

## Hráč

Pokiaľ je hra v režime hry viacerých hráčov, tak sa ostatní hráči budú používatelovi zobrazovať podobne, ako je ukázané na obrázku 3.19. Popis jednotlivých komponentov obrazovky:

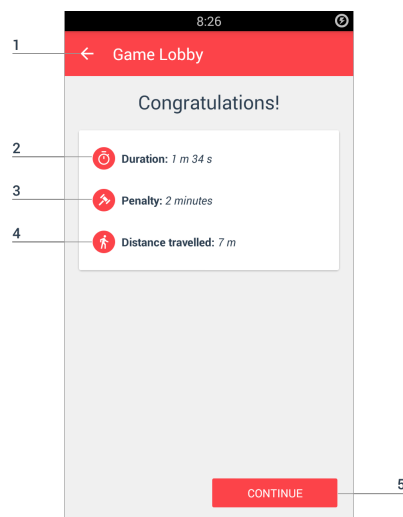
1. Meno hráča. Vykresľovanie sa riadi rovnakými pravidlami ako vykresľovanie názvu objektu cesty.
2. Obrázok hráča.
3. Vzdialenosť od aktuálneho objektu cesty, nie od hráča.



Obr. 3.19: Obrazovka hry, ktorá zobrazuje iného hráča

### 3.5.3 Koniec hry

Obrazovka vyzerá podľa obrázku 3.20.



Obr. 3.20: Obrazovka konca hry

Popis jednotlivých komponentov na obrázku 3.20:

1. Tlačidlo, ktorým sa vrátíme späť na detail cesty.
2. Čas, za ktorý sme prešli cestu.
3. Trestný čas, ktorý sme „získali“ priechodom cesty.
4. Vzdialenosť, ktorú sme prešli.
5. Tlačidlo, ktorým sa vrátíme do modulu prehľadu ciest.

## 4. Programátorská dokumentácia

Táto kapitola popisuje funkčnosť aplikácie z implementačného hľadiska.

### 4.1 Android

Táto sekcia obsahuje popis filozofie operačného systému Android a jeho elementárnych prvkov, ktoré boli použité pri vývoji aplikácie.

#### 4.1.1 Úvod

Android je open source mobilný operačný systém, ktorý je vyvíjaný spoločnosťou Google. Je založený na Linuxovom jadre a je primárne určený pre mobilné zariadenia s dotykovou obrazovkou, akými sú smartfóny a tablety. Avšak v súčasnej dobe sa zameriava aj na iné ciele, ktorými sú televízory, nositeľné zariadenia – tzv. „wearables“ a v neposlednom rade aj autá. Android kladie veľký dôraz na bezpečnosť, preto každá aplikácia žije v jej vlastnom sandboxe a je oddelená od ostatných komponentov. Implicitne aplikácie „skoro“ nič nemôžu a sú potrebné oprávnenia od používateľa. Aplikácie sa primárne píše v jazyku Java, ale je možné ich aj písať v C/C++.

#### 4.1.2 Problémy – roztrieštenosť

Jedným z hlavných problémov operačného systému Android je roztrieštenosť. Tá môže byť:

- **Softwarová** – na trhu je množstvo zariadení s rôznymi verziami Androidu. Je to najmä z toho dôvodu, že výrobcom zariadení sa neoplatí investovať do vývoju softwaru pre staršie zariadenia. Týmto sú vývojari veľmi obmedzovaní a aby mohli cieľiť svoju aplikáciu pre väčšiu skupinu používateľov, tak musia používať zastarané API a používať rôzne support knižnice.
- **Hardwarová** – každé zariadenie je niečím špecifické. Rôznou veľkosťou displeja, rozlíšením, neprítomnosťou rôznych súčastí atď.

Preto nie je možné otestovať aplikáciu na každej konfigurácii, čím môžu vznikáť rôzne problémy.

#### 4.1.3 Základné komponenty aplikácie

Každá aplikácia sa môže skladať zo štyroch základných komponentov, ktorými sú:

- **Aktivity** – predstavujú jednu obrazovku aplikácie a poskytuje možnosť používateľovi zadávať vstup. Typicky sa aplikácia skladá z viacerých aktivít.
- **Services** – sú komponenty, ktoré umožňujú vykonávať dlho bežiacu operáciu na pozadí bez možnosti užívateľskej interakcie.

- **Content providers** – sú komponenty, ktoré umožňujú zdieľať dáta s inými aplikáciami pomocou rozhrania veľmi podobnému práci s databázou.
- **Broadcast receivers** – poskytujú možnosť počúvať na broadcasty, napr. oznámenie o nízkom stave batérie.

Každý komponent musí byť deklarovaný v manifest súbore, ten sa nachádza v koreni projektu pod názvom `AndroidManifest.xml`. Je to xml súbor, ktorý ešte obsahuje:

- **Oprávnenia** – ktoré aplikácia vyžaduje pre svoj plnohodnotný chod. Dôležité najmä pred Android 6, pretože vo verzií 6 sa už na oprávnenia pýta aplikácia priamo za behu.
- **Minimálny API Level** – minimálna verzia Androidu, na ktorom môže daná aplikácia bežať. V novších verziách môže byť prítomné v gradle súbore.
- **Hardwarové a softwarové požiadavky aplikácie** – používatelia, ktorí nespĺňajú dané požiadavky, neuvidia aplikáciu v Google Play obchode.
- **API knižnice, s ktorými musí byť aplikácia zlinkovaná** – napr. Google Maps.

## 4.2 Použité knižnice

Táto sekcia popisuje použité knižnice v projekte a dôležitosť ich využitia.

### 4.2.1 Mobilná aplikácia

Táto časť obsahuje knižnice použité v mobilnej aplikácii.

#### Support knižnice

Tieto knižnice pridávajú mnoho funkcií, ale nie sú súčasťou Android frameworku.

##### v7 appcompat knižnica

Táto knižnica obsahuje podporu materiálového dizajnu a Action Bar-u aj na starších zariadeniach.

##### v13 Support knižnica

Táto knižnica obsahuje najmä triedu `FragmentCompat`, ktorú aplikácia využíva na žiadanie povolení na Android M a novších. Výsledok žiadosti sa potom dostaví priamo do príslušného fragmentu a nie do aktivity.

##### v7 recyclerview knižnica

Táto knižnica obsahuje triedu `RecyclerView`, ktorá je používaná na efektívnejšie zobrazovanie väčšieho množstva dát v zoznamoch.

### **v7 cardview knižnica**

Pridáva podporu pre CardView widget, ktorý je použitý v rôznych častiach aplikácie, napr. ako položka zoznamu ciest atď.

### **Design Support knižnica**

Pridáva najmä podporu Snackbar-ov a FAB<sup>1</sup>-ov.

### **Ostatné**

Ostatné knižnice použité v projekte.

### **Google gson [8]**

Pridáva podporu konverzie medzi Java a JSON objektmi. Funguje obojsmerne. Táto knižnica je vydaná pod Apache licenciou [16].

### **Apache Commons IO [17]**

Pridáva podporu jednoduchej práce so súbormi. Knižnica je taktiež vydaná pod Apache licenciou [16].

### **Google Play Services**

Pridáva podporu najmä pre hru viacerých hráčov, prihlasovanie pomocou Google účtov a prácu s Google mapami.

### **Butter Knife [18]**

Eliminuje najmä zdĺhавé písanie *findViewById(int id)* metód a *onClickListener*-ov pomocou anotácií. Knižnica je vydaná pod Apache licenciou [16].

### **Picasso [19]**

Knižnica umožňuje jednoduché načítavanie obrázkov z URI do *ImageView*. Knižnica je vydaná pod Apache licenciou [16].

### **AESCrypt-Android [20]**

Knižnica pridáva podporu AES šifrovania. Knižnica je vydaná pod Apache licenciou [16].

### **android-crop [21]**

Knižnica pre orezávanie obrázkov. Knižnica je vydaná pod Apache licenciou [16].

### **ColorPickerView [22]**

Knižnica pridáva možnosť vyberať farbu šípky v nastaveniach. Knižnica je vydaná pod Apache licenciou [16].

### **Retrofit [10]**

HTTP klient pre Android, ktorý je využívaný na komunikáciu so serverom. Knižnica je vydaná pod Apache licenciou [16].

---

<sup>1</sup>floating action button

### **Dagger 2 [23]**

Dependency Injection podpora pre Android. Knižnica je vydaná pod Apache licenciou [16].

### **CircleImageView [24]**

ImageView, ktoré zobrazuje obrázok v kruhu. Knižnica je vydaná pod Apache licenciou [16].

### **Google Maps Android API utility library [25]**

Knižnica s nástrojmi pre jednoduchšiu prácu s mapami. Ponúka najmä funkciu na vyrátanie uhla a interpoláciu medzi 2 miestami určenými zemepisnou šírkou a dĺžkou. Knižnica je vydaná pod Apache licenciou [16].

## **4.2.2 Server**

Táto časť obsahuje knižnice použité na serveri.

### **Spring Boot [6]**

Framework, na ktorom je server postavený. Je vydaný pod Apache licenciou [16].

### **H2 [12]**

Databáza servera. Je vydaná pod duálnou licenciou MPL 2.0 [26] a EPL 1.0 [27].

### **Google Api Client Java [28]**

Poskytuje podporu autentizácie používateľa na serveri. Knižnica je vydaná pod Apache licenciou [16].

### **Google Http Client Jackson2 [29]**

Poskytuje súčasti potrebné k správnej funkčnosti autentizácie používateľa. Knižnica je vydaná pod Apache licenciou [16].

## **4.3 Mobilná aplikácia**

Mobilná aplikácia umožňuje používateľovi vytvárať, prechádzať, upravovať a zdieľať cesty.

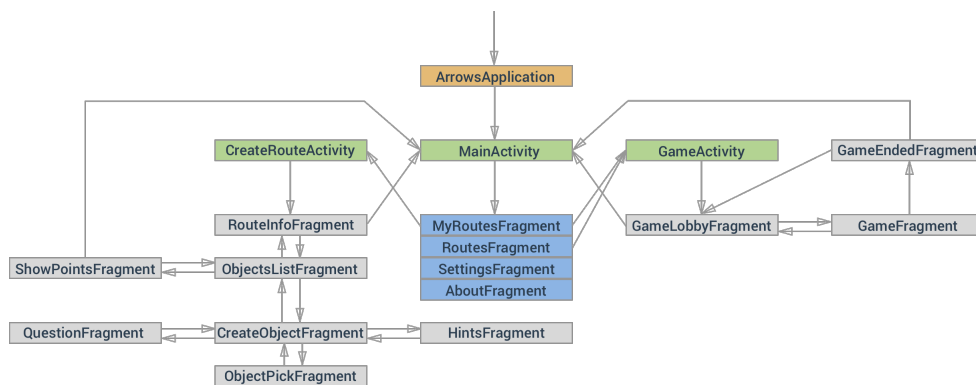
### **4.3.1 Prepojenie UI komponentov**

Na obrázku 4.1 sú zobrazené prechody medzi jednotlivými UI komponentmi. Každý modul je reprezentovaný jednou aktivitou.

Popis jednotlivých komponentov:

- **ArrowsApplication** – komponent, ktorý dedí od *MultiDexApplication* z dôvodu implicitnej podpory pri prekročení limitu počtu metód, ktoré sa môžu nachádzať v jednom DEX súbore. Taktiež pri jej vytvorení inicializujeme Dagger 2 [23] knižnicu, ktorá nám umožní dependency injection v rámci aplikácie.





Obr. 4.1: Prechody medzi komponentami – oranžová farba predstavuje aplikáciu. Zelená farba predstavuje aktivity. Ostatné farby predstavujú fragmenty. Modrou farbou sú označené fragmenty, medzi ktorými sa dá prechádzať pomocou vysúvacieho menu, viz. 3.3.1.

- **MainActivity** – vstupná aktivita aplikácie, ktorá predstavuje modul prehľadu ciest. Táto aktivita obsahuje nasledovné fragmenty:
  - **MyRoutesFragment** – zobrazuje zoznam lokálne uložených ciest na zariadení, viz. 3.3.2.
  - **RoutesFragment** – zobrazuje zoznam ciest uložených na serveri.
  - **SettingsFragment** – zobrazuje nastavenia, ktoré môže používateľ zmeniť.
  - **AboutFragment** – zobrazuje základné informácie o aplikácii.
- **CreateRouteActivity** – slúži ako modul vytvárania cesty. Táto aktivita obsahuje nasledovné fragmenty:
  - **RouteInfoFragment** – obsahuje obrazovku základných informácií o ceste, viz. 3.4.1.
  - **ObjectsListFragment** – obsahuje obrazovku zoznamu vytvorených objektov cesty, viz. 3.4.2.
  - **CreateObjectFragment** – obsahuje obrazovku vytvárania objektu cesty, viz. 3.4.3.
  - **QuestionFragment** – obsahuje obrazovku pridávania otázky a jej odpovedí, viz. 3.4.4.
  - **HintsFragment** – obsahuje obrazovku vyberania nápoved, viz. 3.4.5.
  - **ObjectPickFragment** – obsahuje obrazovky výberu polohy objektu cesty, viz. 3.4.6.
  - **ShowPointsFragment** – obsahuje obrazovku vytvorených objektov cesty, ktoré sa dajú spájať a vytvára sa tým tak cesta, viz. 3.4.7.
- **GameActivity** – slúži ako modul prechádzania cesty. Táto aktivita obsahuje nasledovné fragmenty:
  - **GameLobbyFragment** – obsahuje obrazovku detailu cesty, viz. 3.5.1.

- **GameFragment** – obsahuje obrazovku hry, viz. 3.5.2.
- **GameEndedFragment** – obsahuje obrazovku konca hry, viz. 3.5.3.

### 4.3.2 Perzistencia dát

K vytvoreniu SQLite databázy slúži trieda *DbHelper*, ktorá ako verejné statické atribúty obsahuje názvy tabuliek a príslušných stĺpcov. V metóde *onCreate()* vytvárame tabuľky tak, ako sú popísané v 2.3. Vytvoríme príslušné *UNIQUE*, *NOT NULL* a *FOREIGN KEY* obmedzenia. Taktiež vytvoríme indexy na položky, podľa ktorých sa najčastejšie vyhľadáva. SQLite pre *PRIMARY KEY* indexy automaticky vytvára, ale n rozdiel od iných databáz nevytvára indexy pre *FOREIGN KEY* automaticky, preto si ich musíme vytvoriť.

S databázou pracujeme pomocou triedy *DbUtils*, ktorá nám poskytuje hlavne metódy *getRoute()* a *insertRoute()*, ktorými z databázy získavame cestu a do databázy vkladáme cestu. Aby sme mohli s údajmi pohodlne pracovať, sú vytvorené modelové triedy, ktoré obsahujú atribúty takmer identické s tabuľkami. Tieto modelové triedy sa nachádzajú v balíčku *cz.cuni.mff.arrows.dao.model* a vyzerajú nasledovne:

- **Route** – trieda, ktorá reprezentuje cestu a obsahuje nasledovné atribúty:
  - **long id** – id cesty v lokálnej databáze.
  - **String name** – meno cesty.
  - **int pointsCount** – počet objektov cesty, ktoré cesta obsahuje.
  - **int startingPoint** – relatívne id objektu cesty pre danú cestu.
  - **List<Point> points** – zoznam všetkých bodov cesty.
  - **Bitmap thumbnail** – obrázok cesty.
  - **long remoteId** – id cesty na serveri.
  - **int version** – verzia cesty.
  - **int estimatedTime** – odhadovaný čas prejdienia cesty.
  - **RouteDifficulty difficulty** – enum, ktorý reprezentuje náročnosť cesty.
  - **boolean iAmAuthor** – určuje, či je používateľ autorom cesty. Teda či má používateľ práva meniť cestu.
  - **String description** – popis cesty.
  - **String location** – lokácia štartovného objektu cesty.
- **Point** – trieda, ktorá reprezentuje objekt cesty a obsahuje nasledovné atribúty:
  - **long id** – id objektu cesty.
  - **String name** – meno objektu cesty.
  - **double latitude** – zemepisná šírka.
  - **double longitude** – zemepisná dĺžka.

- **int radius** – rádius objektu cesty.
  - **PointType type** – enum, ktorý predstavuje typ objektu cesty. Možnými hodnotami je šípka, bod a bod s otázkou.
  - **int orientation** – orientácia objektu cesty. Pokiaľ typom nie je šípka, tak táto položka nie je využitá.
  - **Question question** – otázka objektu cesty. Pokiaľ typom je šípka alebo s bodom nie je zviazaná otázka, tak táto položka nie je využitá.
  - **List<Hint> hints** – nápovedy zviazané s daným objektom cesty, ktoré sú usporiadané podľa *order*.
  - **int nextPoint** – predvolený nasledujúci objekt cesty. Pokiaľ je s objektom zviazaná otázka, tak táto položka nie je využitá.
  - **idInRoute** – relatívne id v rámci cesty.
- **Hint** – trieda, ktorá predstavuje nápovedu a obsahuje nasledovné atribúty:
    - **long id** – id nápovedy.
    - **String hint** – text nápovedy.
    - **int penalty** – penalizácia pri zobrazení tejto nápovedy.
    - **int order** – poradie nápovedy.
  - **Question** – trieda, ktorá predstavuje otázku a obsahuje nasledovné atribúty:
    - **long id** – id otázky.
    - **String text** – text otázky.
    - **QuestionType type** – typ otázky.
    - **List<Answer> answers** – zoznam odpovedí.
  - **Answer** – trieda, ktorá predstavuje odpoveď a obsahuje nasledovné atribúty:
    - **long id** – id odpovede.
    - **String answer** – text odpovede.
    - **int nextPoint** – ďalší objekt cesty, pokiaľ si používateľ vyberie túto odpoveď.
    - **int penalty** – penalizácia za túto odpoveď.

### 4.3.3 Získavanie polohy zariadenia

Na získavanie polohy zariadenia využívame triedu *LocationHelper*, ktorá využíva Google Play services location API, ktoré je preferovanou metódou oproti základnému location API v Android frameworku. Základný vyžiadaný interval aktualizácie polohy je nastavený na 10 sekúnd z dôvodu šetrenia batérie za cenu nižšej presnosti. Orientácia sa ale aktualizuje častejšie, jej frekvencia je určená konštantou *SensorManager.SENSOR\_DELAY\_GAME*.

### 4.3.4 Rozšírená realita

Rozšírená realita je implementovaná vo fragmente *GameFragment*. Rozšírená realita sa skladá z 2 *SurfaceView*:

- **MyGLSurfaceView** – je to *GLSurfaceView*, ktorému nastavíme presvitný pixel formát a do ktorého vykresľujeme pomocou inštancie triedy *MyRenderer*. Toto *SurfaceView* sa nachádza nad *SurfaceView* pre kameru, aby sme dosiahli požadovaný výsledok. Správne poradie vykreslenia dosiahneme zavolaním metódy *setZMediaOverlay(true)* na inštanciu *MyGLSurfaceView*.
- **SurfaceView pre kameru** – jeho *SurfaceHolder* predáme ako parameter do inštancie triedy *CameraHelper*, ktorá nám zabezpečí vykresľovanie preview kamery.

Podľa natočenia zariadenia získavame view maticu OpenGL. To dosiahneme tak, že inštanciu triedy *MyRenderer* zaregistrujeme ako listener v inštancii triedy *SensorManager*, čím sa pravidelne volá metóda *onSensorChanged(SensorEvent event)*. V tejto metóde získame rotačnú maticu natočenia zariadenia, ktorú premapujeme tak, ako bolo popísané v 2.2.3. Priamo v tejto matici sa nachádzajú hodnoty vektora pohľadu kamery a vektora, ktorý reprezentuje smer nahor. Hodnoty týchto vektorov môžeme priamo využiť vo volaní *Matrix.setLookAtM(...)*, ktorým dostaneme view maticu, ktorú môžeme ďalej používať pri vykresľovaní.

### 4.3.5 Vykresľovanie

Návrh, princípy a niektoré časti kódu sú prevzaté z [30]. Trieda *VertexArray* slúži na ukladanie polôh vrcholov v natívnom poradí. Prácu so shadermi nám uľahčujú pomocné triedy, ktoré je možné nájsť v balíčku *cz.cuni.mff.arrows.-programs*. Shadery sa nachádzajú v *res/raw* adresári aplikácie.

#### Bod

Bod pre vykresľovanie nám reprezentuje trieda *PointObject*. Chceme, aby k nám boli textúry vždy otočené, preto vždy pred vykresľovaním musíme vykonať rotáciu podľa uhla, ktorý zvierá naša poloha s polohou bodu. Vykresľovanie bodu je zložené z dvoch častí:

- Vykreslenie textúr – vykreslenie červeného bodu a bublinky nad ním, na ktorú sa vykreslí text.
- Vykreslenie textu – názov bodu.

#### Textúry

Pre vykresľovanie textúr nám slúži trieda *TexturedRectangle*, ktorá obsahuje inštanciu *VertexArray* s polohami vrcholov a textúr. S touto triedou úzko spolupracuje *TextureShaderProgram*, ktorý slúži pre prácu s príslušným shaderom.

## Text

Kód pre vykresľovanie textu sa nachádza v triede *Text*. Je to upravená verzia kódu z [31]. Princíp je taký, že vytvoríme textúru všetkých ASCII znakov pomocou android triedy *Canvas* a túto textúru potom používame na vykresľovanie znakov. Hlavnou nevýhodou je, že dokáže vykresliť iba ASCII znaky. Preto pred vykreslením textu sa ho pokúsime vždy obrať o dĺžne, mäkčene a inú diakritiku. Pokiaľ text stále obsahuje neznáme znaky, tak namiesto nich sa vykreslí otáznik.

## Šípka

Šípku pre vykresľovanie nám reprezentuje trieda *ArrowObject*. Táto trieda obsahuje statický atribút *VERTEX\_DATA*, ktorý obsahuje polohy vrcholov a ich normálové vektory získané z *.obj* súboru, v ktorom bola uložená šípka. Na aproximáciu osvetlenia je využité osvetlenie podľa vrchola, pričom svetlo má konštantnú polohu. Počítanie farby vrchola je implementované v shaderi *lighting\_vertex\_shader.glsl*. Šípku natočíme správnym smerom podľa jej orientácie.

## Hráč

Vykresľovanie pozície hráča je veľmi podobné vykresľovaniu bodu, až na ten rozdiel, že namiesto textúry bodu sa využije obrázok používateľa, ktorý získame pomocou inštancie triedy *ImageManager*.

### 4.3.6 Komunikácia so serverom

Pre komunikáciu so serverom slúži knižnica Retrofit 2 [10]. Musíme mať vytvorený vzorový interface *ServerService*, ktorý obsahuje všetky volania na server. Knižnica nám vytvorí implementáciu tohto interfacu. Metódy majú generický návratový parameter typu *Call<Trieda>*, ktoré môžeme asynchrónne zavolať pomocou metódy *enqueue()*, ktorá berie ako parameter callback.

### 4.3.7 Hra

Logika hry je implementovaná v triede *MyRenderer*. Vzdialenosť od objektu cesty získavame pomocou metódy *distanceTo()* a uhol medzi polohou používateľa a objektu cesty pomocou metódy *bearingTo()*, ktoré sú implementované v android triede *android.location.Location*.

## 4.4 Server

Serverová aplikácia obsahuje veľmi málo kódu a väčšinu vecí zabezpečuje framework. Serverí beží na adrese <http://arrowsgame.info:8080>.

## Konfigurácia

Konfigurácia servera sa nachádza v súbore *application.properties*. Možnými nastaveniami sú:

- **images.location** – predstavuje cestu do adresára, kde sú uložené a kam sa majú ukladať obrázky cesty.
- **page.items.count** – predstavuje počet položiek na jednu stránku jedného volania do databázy.
- **multipart.maxFileSize** – reprezentuje najväčšiu veľkosť súboru, ktorú môže server prijať.
- **spring.datasource.url** – určuje cestu, kde sú uložené dáta databázy.

## Perzistencia dát

Na perzistenciu dát je využívané JPA a jeho implementácia pomocou Hibernate. Modelové triedy sa nachádzajú v balíčku *cz.cuni.mff.arrows.dao.model* a sú až na triedu *Route* identické s modelovými triedami popísanými v 4.3.2.

Modelová trieda *Route* obsahuje navyše nasledovné položky:

- **String subject** – reťazec, ktorý identifikuje používateľa.
- **String imagePath** – reťazec, ktorý obsahuje cestu k obrázku cesty. Tento súbor je verejne dostupný na adrese `http://arrowsgame.info:8080/img/imagePath` pre jednoduché načítavanie obrázkov v mobilnej aplikácii pomocou knižnice Picasso [19].
- **boolean temporary** – hodnota, ktorá predstavuje, či sme cestu nahrali iba pre potreby hry viacerých hráčov.
- **Date creationTime** – čas, kedy sme danú cestu vytvorili, slúži najmä pre mazanie starých dočasných ciest.

Ale neobsahuje nasledovné položky, ktoré sú potrebné iba pre mobilnú aplikáciu:

- **Bitmap thumbnail** – obrázok je uložený ako súbor a nie v databáze.
- **long remoteId** – remoteId je obyčajné id na serveri
- **boolean iAmAuthor** – je dôležité iba pre mobilnú aplikáciu.

## Práca s databázou

S databázou sa pracuje pomocou *RouteRepository* interface, v ktorom sa dotazy môžu písať v JPQL<sup>2</sup> alebo v špeciálnom Spring Data JPA formáte.

## Autentizácia používateľa

Na server pošleme zo zariadenia reťazec, ktorý získame zavolaním metódy *getIdToken()* na inštancii triedy *GoogleSignInAccount*. Na serveri už iba tento reťazec overíme pomocou inštancie triedy *GoogleTokenVerifier*.

---

<sup>2</sup>Java Persistence Query Language, [https://en.wikipedia.org/wiki/Java\\_Persistence\\_Query\\_Language](https://en.wikipedia.org/wiki/Java_Persistence_Query_Language)

## Spravovanie požiadavkov

Všetky požiadavky sú spravované v triede *RouteController*. Každý požiadavok je identifikovaný vlastným URL<sup>3</sup>. Toto URL je špecifikované anotáciou *@RequestMapping* na metóde, ktorá tento požiadavok spracováva.

---

<sup>3</sup>Uniform Resource Locator, [https://en.wikipedia.org/wiki/Uniform\\_Resource\\_Locator](https://en.wikipedia.org/wiki/Uniform_Resource_Locator)

# Záver

Práca splňuje zadanie. Mobilná aplikácia implementuje možnosti vytvárania, prechádzania a zdieľania ciest, ktoré sú zložené zo šípiek/bodov, ktoré reprezentujú smer ďalšieho pokračovania. Tieto objekty cesty sa používateľovi zobrazujú pomocou rozšírenej reality. Zdieľanie ciest je umožnené pomocou servera, ktorý bol taktiež implementovaný. Mobilná aplikácia taktiež umožňuje hru viacerých hráčov.

Medzi možné rozšírenia práce patrí história prejdených ciest. S tým spojené rebričky priechodov ciest a zobrazovanie štatistík používateľa. Taktiež možná detailnejšia história prejdenia jednej cesty, ktorá by mohla používateľovi zobrazovať jeho priechod na mape a odpovede pri otázkach. Práca by sa ďalej dala rozšíriť o vyhľadávanie medzi cestami, najmä podľa polohy a mena. Možným rozšírením je už spomenuté webové rozhranie, ktoré by mohlo slúžiť na prehľad ciest, sťahovanie ciest do súborov alebo ich vytváranie vo webovom prehliadači. Ďalším rozšírením by mohlo byť zviazanie obrázkov k odpovediam na otázku. Nakoniec možnosť priradiť k bodu obrázok, ktorý by sa zobrazoval namiesto základnej textúry.



# Zoznam použitej literatúry

- [1] Google. *Google Play Games Services*. URL: <https://developers.google.com/games/services/>.
- [2] Khronos Group. *OpenGL*. URL: <https://www.opengl.org>.
- [3] The MathWorks, Inc. *Android Sensor support from MATLAB*. URL: <http://www.mathworks.com/matlabcentral/fileexchange/40876-android-sensor-support-from-matlab--r2013a--r2013b-/content/sensorgroup/Examples/html/CapturingAzimuthRollPitchExample.html>.
- [4] Google, Inc. *Sensors Overview*. URL: [https://developer.android.com/guide/topics/sensors/sensors\\_overview.html](https://developer.android.com/guide/topics/sensors/sensors_overview.html).
- [5] John Kessenich Bill Licea-Kane Dave Shreiner Graham Sellers. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3*. 8th ed. Addison-Wesley Professional, 2013. ISBN: 978-0321773036.
- [6] Pivotal Software. *Spring Boot*. URL: <http://projects.spring.io/spring-boot/>.
- [7] Zengularity Lightbend. *Play Framework*. URL: <https://www.playframework.com>.
- [8] Google, Inc. *Gson*. URL: <https://github.com/google/gson>.
- [9] *Jackson*. URL: <https://github.com/FasterXML/jackson>.
- [10] Square, Inc. *Retrofit*. URL: <http://square.github.io/retrofit/>.
- [11] The HSQL Development Group. *HSQLDB*. URL: <http://hsqldb.org>.
- [12] Thomas Mueller. *H2*. URL: <http://www.h2database.com/html/main.html>.
- [13] Apache Software Foundation. *Apache Derby*. URL: <https://db.apache.org/derby/>.
- [14] *Performance Comparison*. URL: <http://www.h2database.com/html/performance.html>.
- [15] Red Hat, Inc. *Hibernate ORM*. URL: <http://hibernate.org/orm/>.
- [16] Apache Software Foundation. *Apache license*. URL: <http://www.apache.org/licenses/LICENSE-2.0>.
- [17] Apache Software Foundation. *Apache Commons IO*. URL: <https://commons.apache.org/proper/commons-io/>.
- [18] Jake Wharton. *Butter Knife*. URL: <http://jakewharton.github.io/butterknife/>.
- [19] Square, Inc. *Picasso*. URL: <http://square.github.io/picasso/>.
- [20] Scott Alexander-Bown. *AESCrypt-Android*. URL: <https://github.com/scottyab/AESCrypt-Android>.
- [21] SoundCloud. *android-crop*. URL: <https://github.com/jdamcd/android-crop>.

- [22] Daniel Nilsson. *ColorPickerView*. URL: <https://github.com/danielnilsson9/color-picker-view>.
- [23] Square, Inc., Google, Inc. *Dagger 2*. URL: <https://github.com/google/dagger>.
- [24] Henning Dodenhof. *CircleImageView*. URL: <https://github.com/hdodenhof/CircleImageView>.
- [25] Google, Inc. *Google Maps Android API utility library*. URL: <http://googlemaps.github.io/android-maps-utils/>.
- [26] *Mozilla Public License 2.0*. URL: <https://www.mozilla.org/en-US/MPL/2.0/>.
- [27] *Eclipse Public License 1.0*. URL: <https://opensource.org/licenses/eclipse-1.0.php>.
- [28] Google, Inc. *Google Api Client Java*. URL: <https://developers.google.com/api-client-library/java/>.
- [29] Google, Inc. *Google Http Client Jackson2*. URL: <https://developers.google.com/api-client-library/java/google-http-java-client/>.
- [30] Kevin Brothaler. *OpenGL ES 2 for Android: A Quick-Start Guide*. 1st ed. Pragmatic Bookshelf, 2013. ISBN: 978-1937785345.
- [31] Aleksandar Kodzhabashev. *Texample2*. URL: <https://github.com/d3alek/Texample2>.

# Zoznam použitých obrázkov

|      |  |    |
|------|--|----|
| 2.1  | Zjednodušená interakcia aplikácie so serverom . . . . .                                  | 5  |
| 2.2  | Príklad cesty . . . . .  | 6  |
| 2.3  | Znázornenie tabuliek, ktoré obsahujú údaje cesty . . . . .                               | 9  |
| 2.4  | Interakcia modulov . . . . .   | 12 |
| 2.5  | Zobrazenie hodnôt azimuth, pitch a roll . . . . .  | 20 |
| 2.6  | Nový koordinačný systém zariadenia . . . . .   | 21 |
| 2.7  | Hodnoty azimuthu voči severu . . . . .   | 21 |
| 2.8  | Hodnoty pitch . . . . .  | 21 |
| 2.9  | Obyčajné hodnoty uhla . . . . .  | 22 |
| 2.10 | Príklad vykresľovania pomocou OpenGL . . . . .   | 22 |
| 3.1  | Vysúvacie menu . . . . .   | 29 |
| 3.2  | Vysúvacie menu, pokiaľ je používateľ prihlásený pomocou účtov . . . . .                  | 29 |
| 3.3  | Zoznam lokálne uložených ciest . . . . .   | 30 |
| 3.4  | Obrazovka aktívnej položky zoznamu lokálne uložených ciest . . . . .                     | 31 |
| 3.5  | Obrazovka základných informácií o ceste . . . . .  | 32 |
| 3.6  | Obrazovka zoznamu vytvorených objektov cesty . . . . .                                   | 33 |
| 3.7  | Obrazovka vytvárania objektu cesty . . . . .   | 34 |
| 3.8  | Obrazovka vytvárania otázky . . . . .  | 35 |
| 3.9  | Obrazovka vyberania nápoved . . . . .  | 36 |
| 3.10 | Obrazovka výberu polohy bodu . . . . .   | 37 |
| 3.11 | Obrazovka výberu polohy a orientácie šípky . . . . .                                     | 38 |
| 3.12 | Legenda farieb objektov cesty . . . . .  | 38 |
| 3.13 | Obrazovka vyberania nasledujúcich bodov . . . . .  | 39 |
| 3.14 | Obrazovka výberu nasledujúceho objektu pre bod, s ktorým je<br>zviazaná otázka . . . . . | 40 |
| 3.15 | Obrazovka výberu štartovného objektu cesty . . . . .                                     | 40 |
| 3.16 | Obrazovka detailu cesty . . . . .  | 41 |
| 3.17 | Obrazovka hry, ktorá zobrazuje bod . . . . .   | 42 |
| 3.18 | Obrazovka hry, ktorá zobrazuje šípku . . . . .   | 43 |
| 3.19 | Obrazovka hry, ktorá zobrazuje iného hráča . . . . .                                     | 44 |
| 3.20 | Obrazovka konca hry . . . . .  | 44 |
| 4.1  | Prechody medzi komponentmi . . . . .   | 49 |

# Zoznam použitých skratiek

**AES** Advanced Encryption Standard – *špecifikácia šifrovania elektronických dát*

**API** Application Programming Interface – *sada pravidiel a špecifikácií, ktoré programy implementujú pre správnu komunikáciu medzi sebou*

**CRUD** Create, read, update and delete – *skratka pre vytváranie, čítanie, aktualizovanie a mazanie*

**DEX** Dalvik EXecutable – *kompilovaný balík kódu v systéme Android*

**HTTP** Hypertext Transfer Protocol – *protokol pre prenos html dokumentov*

**JPA** Java Persistence API – *špecifikácia, ktorá popisuje manažment relačných dát v Java aplikáciách*

**JPQL** Java Persistence Query Language – *objektovo orientovaný dotazovací jazyk*

**JSON** JavaScript Object Notation – *jazykovo nezávislý formát dát*

**OpenGL** Open Graphics Library – *API pre tvorbu počítačovej grafiky*

**ORM** Object-relational mapping – *technika, pomocou ktorej pracujeme s objektmi namiesto s relačnou databázou*

**SOAP** Simple Object Access Protocol – *protokol na výmenu správ založených na XML*

**SQL** Structured Query Language – *špeciálny programovací jazyk navrhnutý pre prácu s relačnými databázami*

**URI** Uniform Resource Identifier – *reťazec znakov používaný na identifikáciu alebo pomenovanie zdroja*

**XML** eXtensible Markup Language – *jazykovo nezávislý formát dát*

# Prílohy

## Príloha A

Obsah priloženého CD je nasledovný:

1. **Dokumentácia** – dokumentácia pre mobilnú aplikáciu vygenerovaná pomocou programu *javadoc* sa nachádza v adresári *documentation*.
2. **Zdrojové kódy** – projekty sú umiestnené v nasledujúcich adresároch:
  - **Mobilná aplikácia** – *projects/app*
  - **Server** – *projects/server*
3. **Skompilovaný inštalačný balík aplikácie** – nachádza sa v súbore *bin/arrows.apk*.